AD/A-004 295

FAILURE ANALYSIS BY STATISTICAL TECHNIQUES (FAST). VOLUME I. USER'S MANUAL

William H. Rowan, et al

TRW Systems Group

Prepared for:

Defense Nuclear Agency

31 October 1974

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

Best Available Copy

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DNA 3336F-1	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER AD/A-004295
4. TITLE (and Subtitle) FAILURE ANALYSIS BY STATISTICAL TECHNIQUES (FAST) Volume I, User's Manual		5. TYPE OF REPORT & PERIOD COVERED Final Report for Period 1 April 1973–30 April 1974
		6. PERFORMING ORG. REPORT NUMBER 24006-001-RU-00
7. AUTHOR(s) William H. Rowan, Norman Lipner Jack J. Farrell, John D. Oliver Aileen S. Nii		8. CONTRACT OR GRANT NUMBER(s) DNA 001-73-C-0222
9. PERFORMING ORGANIZATION NAME AND ADDRESS TRW Systems Group One Space Park Redondo Beach, California 90278		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NWED Subtask Y99QAXSC061-21
11. CONTROLLING OFFICE NAME AND ADDRESS Director Defense Nuclear Agency Washington, D.C. 20305		12. REPORT DATE 31 October 1974
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This work sponsored by the Defense Nuclear Agency under Subtask Y99QAXSC061-21.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

FAST

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This document is a user's manual for the FAST (Failure Analysis by Statistical Techniques) computer code. FAST is a technique for performing a statistical evaluation of the survivability of military or civilian systems subjected to a hostile environment such as would be caused by a nuclear burst or earthquake. The document contains a description of the analysis formulation, the inputs needed to perform system assessment studies, and the code and its operation, and a demonstration of the code operation through documentation of a sample

20. ABSTRACT (Continued)

problem which has been designed to address the key aspects of the methodology and its practical application. The two volumes of the document are those of Volume I, User's Manual, and Volume II, User's Manual (Appendices). The code itself was developed by TRW Systems Group over the past decade, mainly for MINUTEMAN under sponsorship of the Air Force Space and Missile Systems Organization (SAMSO).

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF TABLES

FAILURE ANALYSIS BY STATISTICAL TECHNIQUES (FAST)

1.0 INTRODUCTION

FAST is a tool developed over the past decade because of the need for evaluating the nuclear survivability of strategic weapons systems. These weapons systems happened to employ many sites built almost identically. This led to early recognition that a statistical approach would be useful - to take into account such random variations as those in soil properties and construction quality.

The FAST methodology has been applied mostly to in-place weapons systems for the purpose of evaluating inherent system hardness or the benefit of hardness improvements. The FAST technique has also been used in safety analysis of ships, and is applicable to earthquake design. In general, it has wide potential applicability to the hardness/survivability evaluation of any military or civilian system. The purpose of this document is to transmit the FAST capability to other potential users so they can employ it in their applications.

This report contains the following information needed to enable a prospective user to understand and apply the FAST techniques:

- A description of the analysis formulation and of the inputs needed to perform system assessment studies
- Documentation of the computer code and its operation
- Demonstration of the code operation through documentation of a sample problem which has been designed to address the key aspects of the methodology and its practical application.

2.0 FAST METHODOLOGY

This section describes the methodology of the FAST technique, the analysis formulation and system modeling considerations.

2.1 FAST CONCEPT

The hardness evaluation or assessment of a complex system subjected to a hostile environment requires calculation of the probability of response of each component to the hostile environment, determination of the probability of failure of each component for that response, and the combination of the component probabilities to obtain the failure probability of the system. The FAST code has been designed to perform this evaluation in a manner such that parametric sensitivity, trade-off and optimization studies can be readily accomplished.

Typically, all system components are identified and catalogued. In order to more conveniently catalogue the system, its components can be grouped into subsystems. Then the possibility of each component or subsystem directly or indirectly contributing to system failure is ascertained. The individual failure mechanisms and fragilities of the components are determined and related to parameters of the hostile environments.

A diagram describing the FAST methodology is shown in Figure 2-1. The figure illustrates how, for a simplified system, selected parameters of the total hostile environment may be critical to the survivability of a hardened system. The free field environments are transformed by transfer functions to establish local system responses to the environments, which are in turn used to predict component failure probabilities.

In the FAST code, each component failure probability is modeled by a fragility curve that defines for each component or subsystem the probability of failure as a function of the local system response to the free field environments. Component probabilities of failure are combined in system network equations to compute system probability of survivability. The system network is a functional description

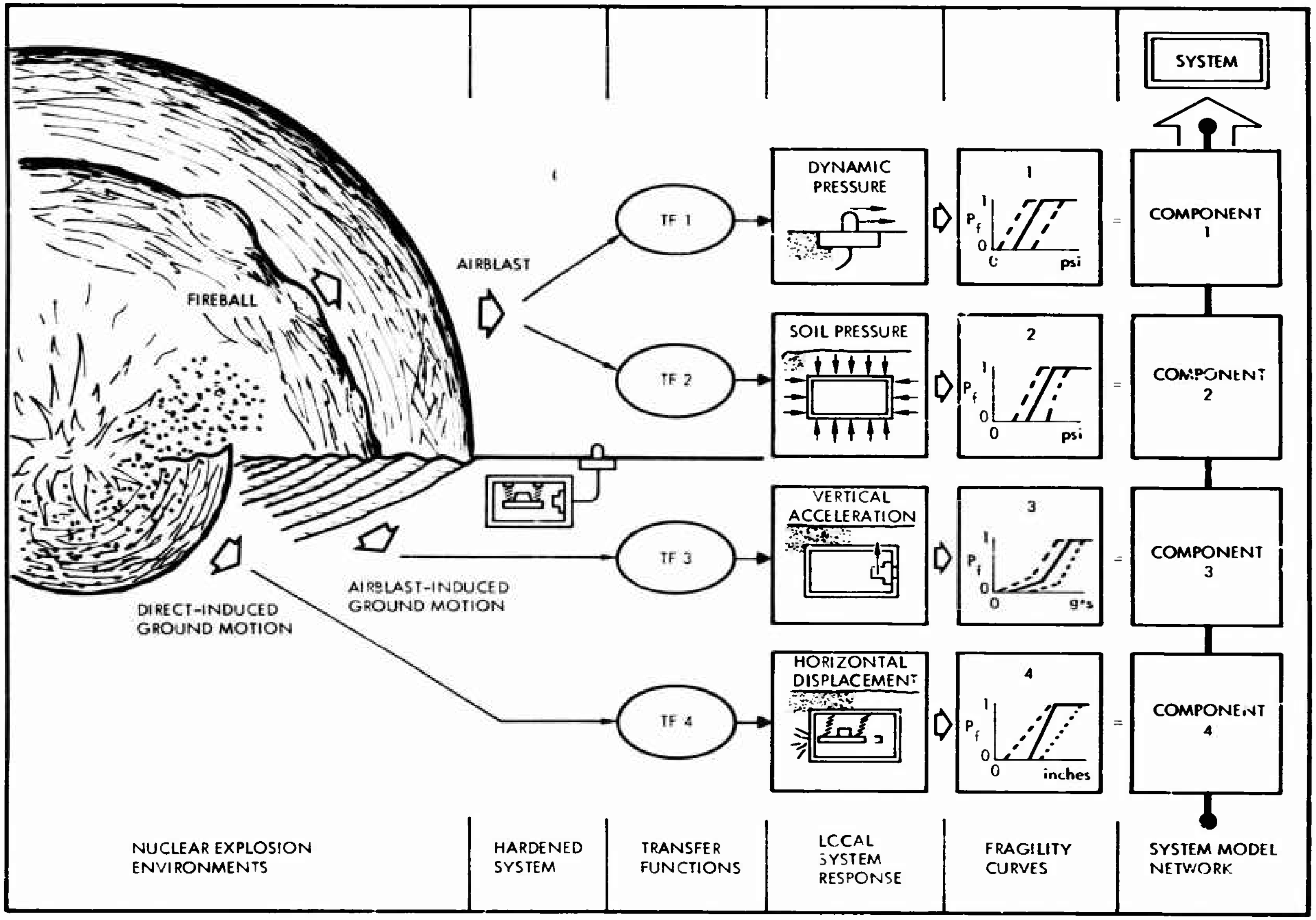


Figure 2-1. FAST Methodology Overview

which specifies the series/parallel relationship between components. Components are in series if all components are required to accomplish an essential system function, and are in parallel if any one could perform the essential function.

An important facet of the FAST methodology is treatment of the underlying uncertainties in predicting system hardness. Of the four system/environment inputs to FAST, the system network is the only one that must be known exactly. The code acknowledges and accommodates finite uncertainty in modeling environments, transfer functions and component fragilities. Uncertainty in environment estimates is often due to the lack of adequate analytic or empirical models for scaling nuclear weapon effects. This is also true of the uncertainties ascribed to transfer functions. Uncertainties in component fragilities are primarily a consequence of insufficient test data on the components at levels near and beyond failure. One of the most valuable features of FAST is that these uncertainties are individually modeled and are properly accounted for in the calculation of system failure probability. Two categories of variations are recognized by the FAST code, namely, random and systematic variations.

The fundamental difference between random and systematic variations is that a systematic variation extends uniformly over a population of facilities whereas random variations extend non-uniformly from one facility to another. Random variations tend to average out over a large population whereas systematic variations do not. It is noted that systematic variations are usually reducible by test and/or analysis programs which improve models of the phenomenology and system behavior. Random variations may also be reduced under some conditions but the value of such improvements is questionable for systems with large populations. The FAST code treats both random and systematic variations of environments, fragilities, and transfer functions. The specific treatment of variation and correlation of FAST parameters is noted in Table 2-A.

Table 2-A. Variations and Correlations of FAST Parameters

Parameter	Random		Systematic	
	Variation	Correlation	Variation	Correlation
Environments	Yes	Yes	Yes	Yes
Transfer Functions	Yes	Yes	Yes	No
Fragilities	Yes	No	Yes	No

Another important feature of FAST is that the code accounts for correlation or covariance between variables. For example, if two components are affected by strongly correlated environments, they will either both tend to fail or survive with the probability of both surviving being higher than if the environments are uncorrelated. Alternatively, if the environments are anti-correlated, the probability of survival of the system can be substantially less than for the uncorrelated case. Relatively strong correlation or anti-correlation between environments is not unusual. For example, a specific system might be vulnerable to a combination of air induced ground shock, direct induced ground shock and debris. Each of these is strongly affected by soil stiffness and therefore is anti-correlated, correlated and anti-correlated, respectively, with soil stiffness. FAST treats correlations between both systematic and random variations of environment parameters. Correlations between fragility parameters and transfer functions are not treated except for azimuth sensitivity of transfer functions, although such correlation is easily incorporated into the code, and has been incorporated in some versions of FAST. The lack of use of these options has led to their deletion from contemporary versions of the code.

The FAST approach combines separate estimates of the environments, transfer functions, fragilities, corresponding correlation and systematic variability estimates, and a system functional network to provide probabilities of survival of the system, subsystems, and components. Probability of survival is given as a best estimate value along with certainty (or confidence) bands for any pre-selected range from a natural or manmade energy source.

The FAST code is standardized for systems deployed in large numbers and built almost identically. The survivability calculations are performed in a fashion to discriminate between random and systematic variations in the survivability statistics. This is accomplished by averaging the survivability statistics over the random variations to suppress this kind of variation (but account for non-linearities), while preserving and displaying the variation in the survivability statistics due to systematic variations in the inputs. However, FAST in its current form can also consider applications involving survivability of systems employing only one or a few sites, if the random and systematic variations can be combined.

2.2 FAST FORMULATION

The general methodology of the FAST technique allows for quite flexible descriptions of parameters; however, the existing code has very specific forms of the environments, transfer functions, fragilities, and system networks which have been found to be the most useful form for code application by the user. Specifically, the analysis has been developed for convenience of application to proliferated systems hardened against nuclear weapons effects. However, parameters can be given more general formulation should this be warranted by other applications.

The computational processes built into the current code are not the only ones that could be used for the FAST technique; however, they have allowed for the most general employment of the FAST concept compared with other computational processes used in the past.

2.2.1 Environments

As previously discussed, the failure probabilities for the component fragilities are derived from free field hostile environments through transfer functions. The FAST hostile environment modeling is quite general. For example, environments may be variables which can directly cause failure of sensitive components, such as ground shock, air blast or debris parameters. Alternatively, they may be variables which indirectly contribute to failure, such as depth to water table if failure might be caused by flooding after some structural element has failed.

At a reference distance from a specific nuclear explosion, a set of nominal values for the free field environments is defined. These nominal values are not the exact values that would occur for each explosion if a large number of explosions were monitored. The variations from the nominal value are both random variations, which cover factors like topography variations at the reference distance from a system, or geology differences from one site to another, and systematic variations, which deal with things such as uncertainty about the cratering efficiency of any particular bomb in a specific geology and our uncertainty in defining the specific geology at the sites of interest. A set of nominal values may be calculated by predicting a set of best estimate environments for each site in the group (or each site/azimuth) and then statistically calculating the mean value for each environment in the set:

$$E_i = \frac{1}{n} \sum_{j=1}^{n} (\epsilon_{ij}). \qquad (2\text{-}1)$$

This set of nominal values is referred to as the "vector of environment means" and is identified in FAST as ULN. These are of course related to specific reference threat conditions. An example giving median environments for a 600 psi overpressure pulse from a 5 MT weapon follows.

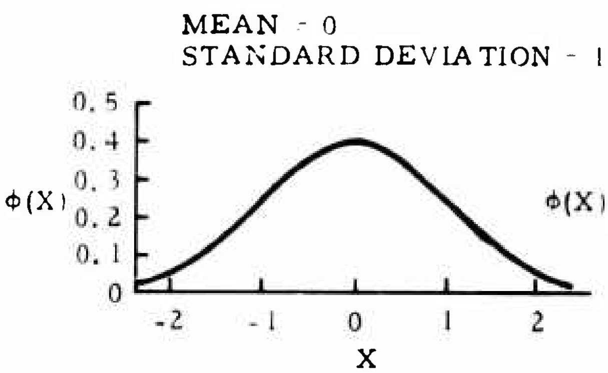
Each environment parameter is represented statistically to take into account the random variations, and systematic variations, as noted in Section 2.1. The normal or log-normal distribution is used to model each environment. An example of the normal distribution is shown in the following sketch. In the log-normal distribution, the logarithms of the parameters are distributed normally. The log-normal distribution for a single variable is given by

$$\phi(\log x) = \frac{1}{\sqrt{2\pi}\, x\, \sigma(\log x)} \exp\left\{ -\frac{1}{2}\left(\frac{\log x - \log \bar{x}}{\sigma(\log x)}\right)^2 \right\} \qquad (2\text{-}2)$$

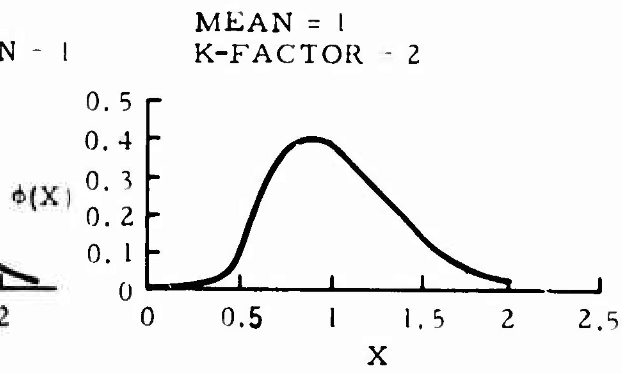
FAST Code		
Parameter	**Designation**	**Example**
Reference Threat	NOM RWH	
Range	R_o	3000 feet
Yield	W_o	5 MT
Height of Burst	H_o	3 feet
Vector of Means	ULN	
Air Induced		
Displacement	E_{01}	15 inches
Velocity	E_{02}	380 in. /sec
Acceleration	E_{03}	60 g's
Direct Induced		
Displacement	E_{04}	27 inches
Velocity	E_{05}	29 in. /sec
Debris	E_{06}	12 inches
Water Table	E_{07}	50 feet

Note: The overpressure/range relationship is from Ref. (3).

where x is the environment parameter, $\sigma(\log x)$ is the standard deviation and ϕ is the probability density. An example of the log-normal distribution is shown in the following sketch. Frequently, strictly positive random variables follow the log-normal distribution which has proved to be adequate for all past calculations; however, the distributions could be expanded to include other density functions.



NORMAL DISTRIBUTION



LOG-NORMAL DISTRIBUTION

As discussed in Section 2.1, the variability of different environments is not usually independent, so that correlations between environments must be properly modeled in the code. Correlation is modeled by correlation coefficients which mathematically are defined by

$$\rho_{xy} = \frac{\mathrm{Cov}\,(x,y)}{\sigma_x \sigma_y} \tag{2-3}$$

where σ_x and σ_y are the standard deviations of the random variables x and y, and Cov (x, y) is defined as

$$\mathrm{Cov}\,(x,y) = \iint (x-\bar{x})\,(y-\bar{y})\,f(x,y)\,dxdy \tag{2-4}$$

The quantities $\bar{x}$ and $\bar{y}$ are the mean values of x and y, and f (x, y) is the joint probability density function. When ρ_{xy} is zero the random variables are linearly independent; when ρ_{xy} is unity the random variables are linearly correlated; and where ρ_{xy} is minus one the random variables are linearly anti-correlated. For multivariate distributions, the variances and correlation coefficients can be expressed in matrix form as shown.

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \text{-----} & \rho_{1n}\sigma_1\sigma_n \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 & \text{-----} & \rho_{2n}\sigma_2\sigma_n \\ \vdots & & & \vdots \\ \rho_{1n}\sigma_1\sigma_n & & \text{-----} & \sigma_n^2 \end{bmatrix} \tag{2-5}$$

This array is known as the covariance matrix. The σ's are the standard deviations of the environments (linear or log) so that the matrix can be used to define the joint probability density function for all the environments as shown,

$$\phi(x_1, x_2, \text{--}, x_n) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}\widetilde{(x-\bar{x})}\Sigma^{-1}(x-\bar{x})\right\} \tag{2-6}$$

where $(x - \bar{x})$ is an n-component column vector comprising all of the environment parameters, $\widetilde{(x - \bar{x})}$ is its transpose, Σ^{-1} is the inverse of the covariance matrix and $|\Sigma|$ is the determinant of Σ. It is noted that some of the variables in this distribution may be logarithms of environment parameters, while others are the environment parameters themselves. The only requirement is that the distribution of the variable, or its logarithm, be normal. References 1 and 2 contain more detailed information on the properties of multivariate and log-normal distributions. Two covariance matrices are modeled for each FAST calculation. One corresponds to the random variation, and the other to the systematic variation. The former is input as the S-matrix and can be derived from best-estimate calculations for each siting area (or azimuth/site group). The second is described by a K-factor vector, defining the systematic variation in each environmental estimate, and an A-matrix characterizing the correlation of these variations (see Section 3).

These covariance matrices are not scaled; however, individual matrices can be specified for each threat condition. Experience indicates that the matrices for weapon environment parameters are at least approximately constant if the scaling laws are linear (in their logarithmic form).

The multivariate normal distribution is defined by its mean vector, $\bar{x}$, and its covariance matrix, Σ. The role of each of these in FAST analysis will now be discussed.

Median environment parameters are first defined for a reference threat condition. If the analysis concerns a facility subjected to nuclear attack, the median environment parameters are defined for a baseline threat condition of yield (W_o), miss distance range (R_o) or corresponding mean overpressure (P_o), and height of burst (H_o). These environments

are then scaled to the other threat condition of interest (W, R or P and H), using either of the two forms of scaling laws shown

$$E = E_o \left(\frac{W}{W_o}\right)^{\alpha} \left(\frac{R}{R_o}\right)^{\beta} \exp\left\{-\gamma\left[\frac{H}{W^{1/3}} - \left(\frac{H}{W^{1/3}}\right)_o\right]\right\} \quad (2\text{-}7a)$$

or

$$E = E_o \left(\frac{W}{W_o}\right)^{\alpha} \left(\frac{P}{P_o}\right)^{\beta} \exp\left\{-\gamma\left[\frac{H}{W^{1/3}} - \left(\frac{H}{W^{1/3}}\right)_o\right]\right\} \quad (2\text{-}7b)$$

where α, β and γ are scaling factors, and E_o and E are the baseline and scaled environment parameters, respectively.

The FAST code contains the Brode overpressure relationship with height of burst and range, specified in Reference 3. Through this equivalence between nominal overpressure and range, the choice of scaling law for each environment parameter is purely a matter of convenience.

The height of burst scaling is designed to model crater-related effects for near surface burst conditions. A much more complex scaling would be needed to cover all conditions from surface to above optimum height of burst. High altitude burst conditions can be considered by suppressing the height of burst scaling and letting the baseline threat contain the height of burst of interest.

2.2.2 Transfer Functions

Transfer functions relate the free field environments to the local system response of components. In normal usage a transfer function will take a single environment (for example overpressure) and convert it into a response which governs a system failure mode (for example shear force in a facility roof). The transfer function can be used to account for structural dynamics, impedance mismatch, impulse effects, etc. Moreover, transfer functions can be used in ways which simplify,

clarify, or replace environment generation. Indeed, environments, transfer functions and fragilities are so uniquely correlated that any particular detail of the modeling might be incorporated in any one as well as the other.

Transfer functions can be a function of the attack azimuth (θ). Problems where applied loads vary with azimuth, due to non-axisymmetric hardware or site conditions, can be modeled in this manner. In its simplest form, a transfer function (TF) relates an environment parameter (E) to a response (R), viz.;

$$R = E \times TF(\theta) \tag{2-8}$$

A transfer function may be less than unity, representing attenuation, or it may be greater, representing amplification.

Transfer functions are represented statistically in the FAST code. Systematic variation is assumed to have a beta distribution specified by a K-factor, with 95% of the distribution falling between (median/K) and (K x median) and 2-1/2% at each end. The code selects the best fit from the nine choices of the beta distributions shown in Figure 2-2. This table can be replaced to provide a different family of distributions if such is appropriate to a given application. As discussed in 2.2.3, the same table is used for fragility systematic variation.

Random variation of transfer functions is introduced into the FAST code to account for azimuth sensitivity of components in the analysis of facilities subjected to nuclear attack. This is accomplished by use of azimuth sensitivity tables. An azimuth is selected at random (uniform distribution) and used for all components.

In the more general form of the transfer function, a single system response can be written as a linear combination of up to 30 environment parameters, viz.;

$$R = E_1 \times TF_1(\theta) + E_2 \times TF_2(\theta) + \ldots \tag{2-9}$$

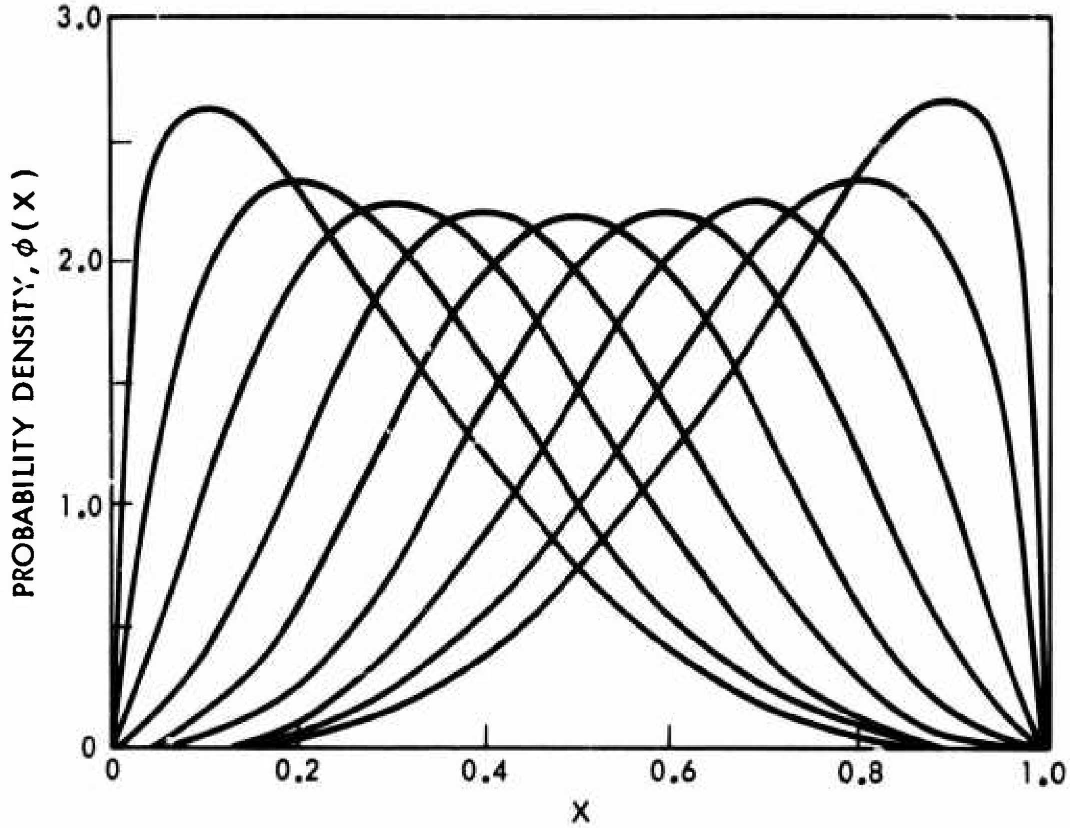


Figure 2-2. Beta Distribution Shapes

Such a formulation is very useful for modeling dynamic response when modal participation is a factor.

2.2.3 Fragilities

A fragility defines the probability of failure of a component as a function of a local system response (frequently a critical local environment). In many applications, failure is assumed to occur if the component experiences damage sufficient to prevent (by itself or in combination with other failures) successful completion of the system mission. No other damage, no matter how severe, should be considered failure. However, the code is simply a tool, and if another failure definition is more appropriate for a specific application, it could of course be used.

Possible forms of fragilities are shown in Figure 2-3. The ordinate is probability of failure, ranging from 0 to 1, and the abscissa is a component response (ordinarily a critical local environment) which can cause failure. A component may have more than one fragility, which might correspond to different failure modes or other system modeling considerations. Each fragility must represent the best engineering estimate of the actual failure probability function, not a conservative estimate. It must represent as much likelihood of being too low as being too high. All fragilities are treated as being uncorrelated.

A fragility curve is a cumulative probability function which, in general, is sloped due to postulated differences among a group of like components or other random variations. For example, individual components built to the same drawings may fail at different response levels because of unintentional differences in tolerance, material properties, parts production variation, welding, manufacturing procedures, maintenance, operating environment, or other factors. Furthermore, fragility testing considers only an approximation of the local environment, while fragility analyses consider only an approximate model of the component. The resulting uncertainties lead to both random and systematic variations.

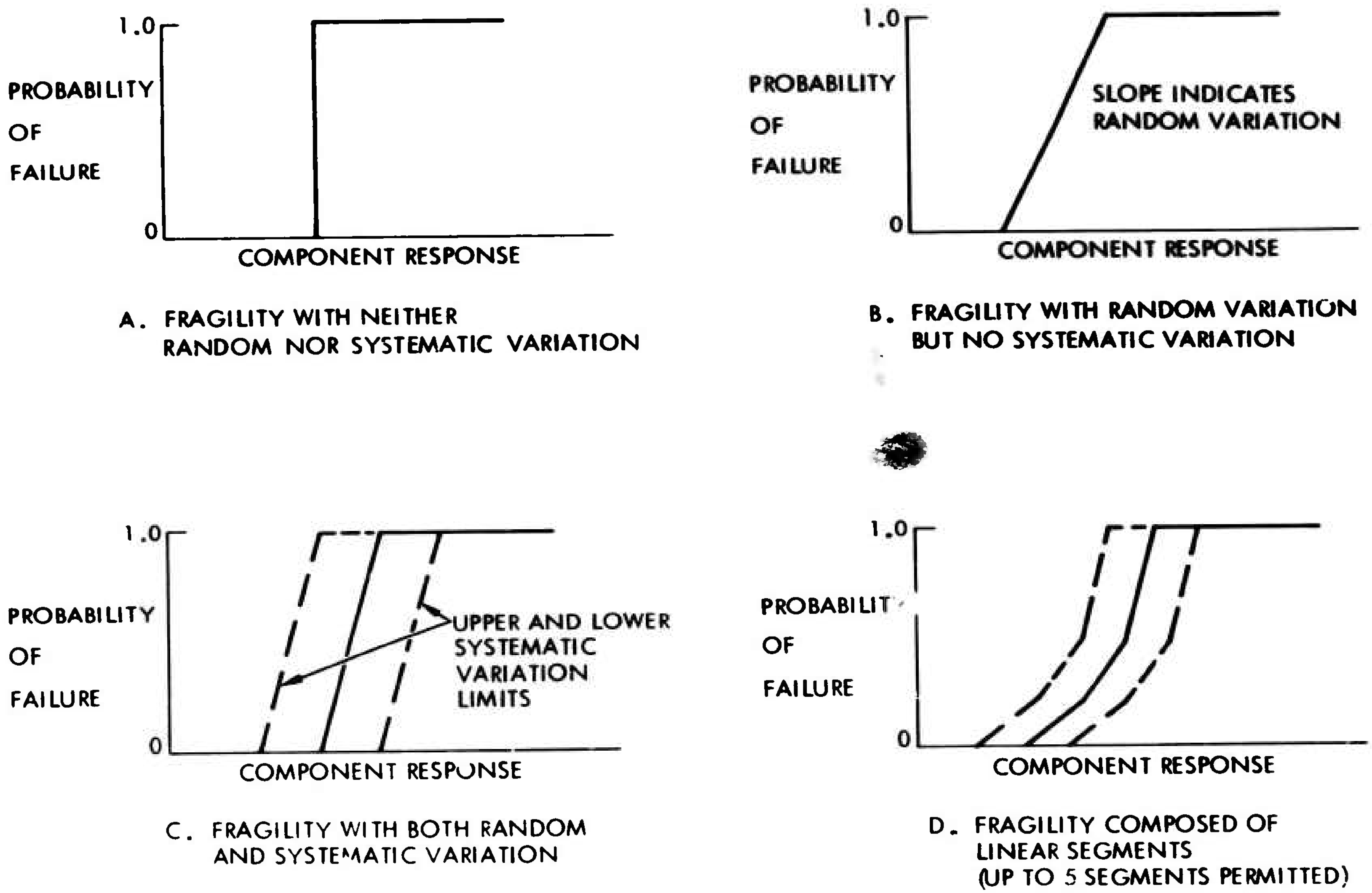


Figure 2-3. Examples of Fragilities

Figure 2-3A shows a cookie cutter fragility, i.e., without random variation; whereas 2-3B shows a fragility with random variation. Analysis and/or specific tests may be used to determine the steepness and/or straightness of the curve. FAST allows the general fragility curve to be composed of, at most, five linear segments beginning at zero and ending at one. The format is illustrated in Figure 2-3D.

The upper and lower dashed boundaries in Figures 2-3C and 2-3D reflect the systematic variation or uncertainty inherent in establishing the best estimate curves. The boundary lines, in general, need not be parallel or symmetric about the nominal curve.

The fragility systematic variation within the uncertainty band is assumed to have a beta distribution with the best estimate centered at the mode of the distribution. The code selects the best fit from the nine choices of the beta distribution shown in Figure 2-2.

Azimuth sensitivity of fragilities can also be accounted for in FAST. The simplest example of azimuthal effects on fragilities is a shock-isolated component with different amounts of rattlespace in different directions. One assigns a reference direction for the problem and all azimuth sensitivities are measured from the reference azimuth. Then during calculation, random azimuths are selected, and for each azimuth all environment parameters are converted to fragility responses, taking into account azimuth sensitivity, and compared with critical fragility levels. In this way systems which have the same weak directions for multiple components will not have multiple kills traceable to weak direction effects, when all the azimuthal failures occur either concurrently or not at all.

2.2.4 System Network

The logic enabling the code to combine component failure probabilities into subsystem and system failure probabilities solves probability equations based upon the system network. The code uses two levels of networks: components are combined to form subsystems and subsystems and/or components are combined to form systems. Figure 2-4 shows several examples of such networks: 2-4A is a subsystem of three components in series, 2-4B a subsystem of three

components in parallel and 2-4C is a subsystem including both series and parallel components. Systems are defined as combinations of subsystems and/or components in an analogous way, as indicated by Figure 2-4D.

For components in series, a failure of any one of them will cause subsystem failure. Thus, if the probabilities of failure for the components in Subsystem 1 of Figure 2-4 are P_{101}, P_{102} and P_{103} then the probability that this subsystem fails is

$$P_{SS1} = 1 - (1-P_{101})(1-P_{102})(1-P_{103}) \qquad (2\text{-}10a)$$

For components in parallel, a failure of all is required for subsystem failure. Thus, the probability of failure for Subsystem 2 is

$$P_{SS2} = P_{201} \cdot P_{202} \cdot P_{203} \qquad (2\text{-}10b)$$

Any subsystem or system can be modeled as combinations of series and parallel elements. Consequently, the failure probabilities for the other networks of Figure 2-4 are given by

$$\begin{aligned} P_{SS3} = 1- \{1-\left[1-(1-P_{302}\, P_{303})(1-P_{301})\right] \\ \left[1-(1-P_{304})(1-P_{305})\right]\}(1-P_{306}) \end{aligned} \qquad (2\text{-}10c)$$

$$P_{SYS1} = 1- \left(1-P_{SS3}\, P_{SS4}\right)(1-P_{SS1})(1-P_{SS2}) \qquad (2\text{-}10d)$$

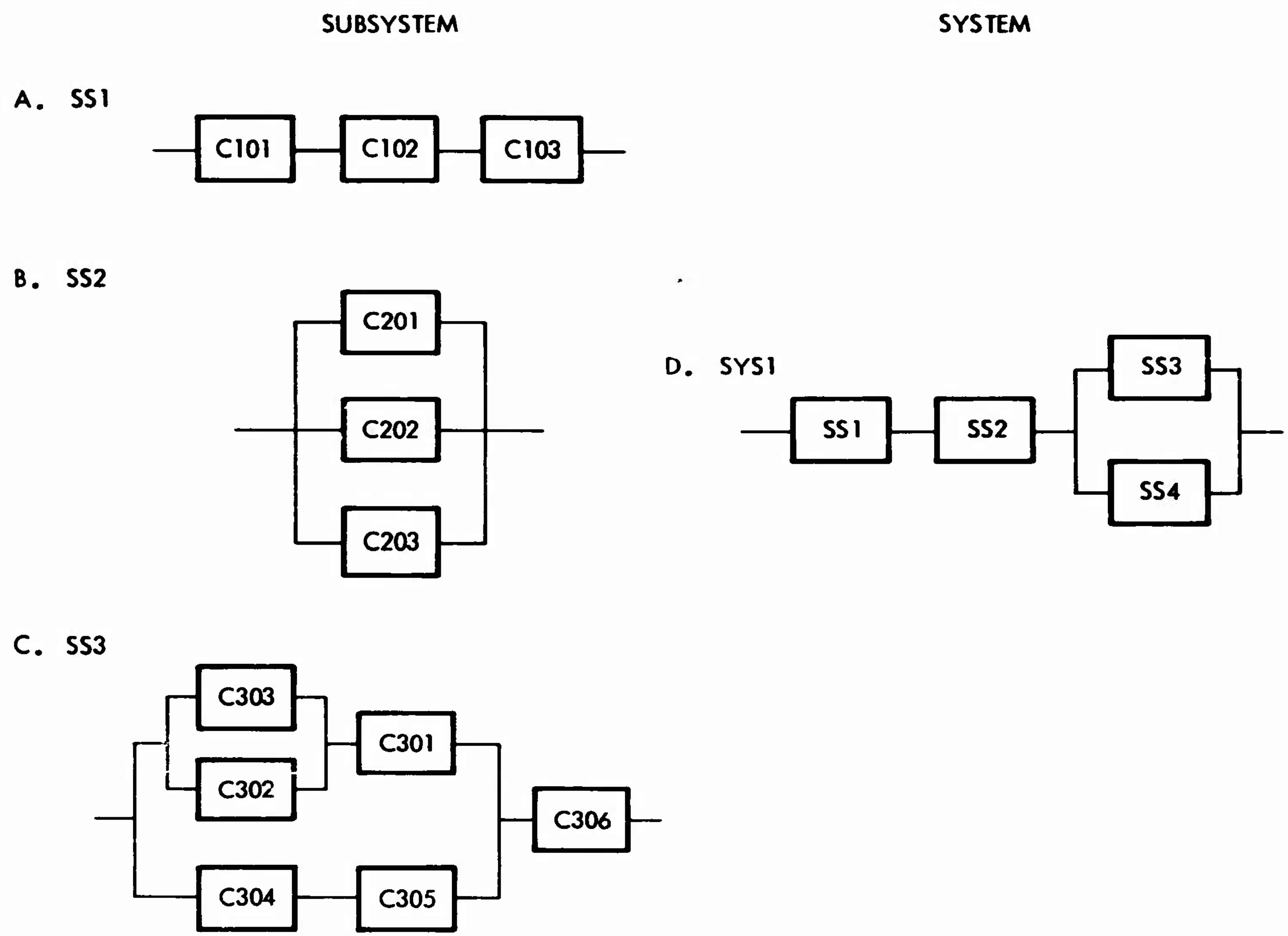


Figure 2-4. Examples of Subsystem and System Networks

Derivation and programming of such mathematical expressions is not practical for complex systems, so the FAST code evolution has developed a simplified automated technique for this purpose. The technique employs software called the Boolean Network Compiler. The relationship between the various components and subsystems is input, and the compiler sets up the correct mathematical operations to compute subsystem and system survivability. To illustrate the simplicity of this technique, equations 2-10a through 2-10d would be input as follows:

$$SS1 = C101 + C102 + C103 \tag{2-10a'}$$

$$SS2 = C201 * C202 * C203 \tag{2-10b'}$$

$$SS3 = (C302 * C303 + C301) * (C304 + C305) + C306 \tag{2-10c'}$$

$$SYS1 = SS1 + SS2 + SS3 * SS4 \tag{2-10d'}$$

The Boolean operators + and * denote the series and parallel relationships, respectively.

The ability to name several systems in a single run by varying one or more subsystems or components is called "ganging", and is useful for parametric studies. This is because once the component failure probabilities have been computed, they can be combined in different ways according to the network equations to obtain various system failure probabilities.

2.2.5 Survivability Statistics

This section describes how the FAST code combines the statistical representations of environments, fragilities, and transfer functions in accordance with the system network definitions to obtain a statistical description of the system survivability.

2.2.5.1 Survivability Calculation

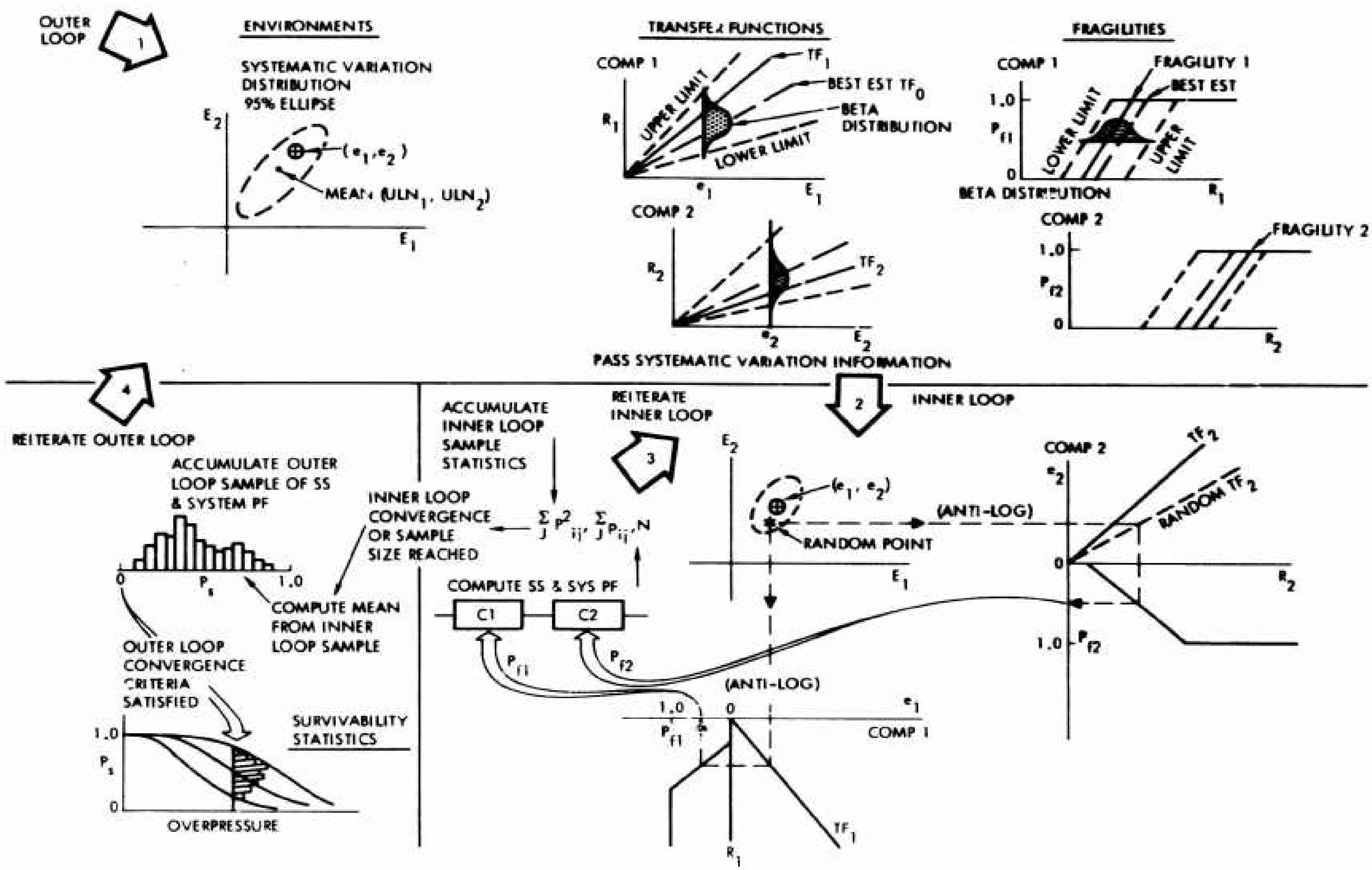
The FAST code is standardized for proliferated hardened systems as discussed in Section 2.1. Random variables affecting system survivability are appropriately averaged by the process, while the systematic variations are propagated through the calculations to indicate the level of certainty (or confidence) in the system survivability result.

The Monte Carlo technique is used in performing the survivability calculations. This technique samples from the random and systematic distributions of the environments, fragilities, and transfer functions to obtain values used to compute a system survival probability. This process is repeated many times to obtain a sample of survival probabilities which can be summarized and displayed as system survivability statistics.

An overview of the computation process is diagrammed in Figure 2-5 for a simplified system of just two components. These components, together with their associated environments, fragilities, and transfer functions, are denoted by the subscripts 1 and 2. The FAST code accomplishes the sampling process in two stages, to discriminate between random and systematic variations.

The first stage, accomplished in an outer loop of the code, selects bias values from the systematic variation distributions for the environments, fragilities, and transfer functions. This is illustrated by the point ⊕ (e_1, e_2), selected from the environment systematic variation distribution in the upper left in Figure 2-5, by the selection of the solid lines labeled Biased TF_1 and Biased TF_2 from the transfer function systematic variation distributions and by selection of the solid lines labeled Fragility 1 and Fragility 2 from the fragility systematic variation distributions.

The systematic variation information defined in the outer loop is passed to the inner loop where the second stage of the sampling process is accomplished. Values are selected from the random variation distributions for the environments and transfer functions (azimuth sensitivity). The environment values are chosen from the multivariate normal distribution defined by the mean ((e_1, e_2) in the illustration) and the random covariance matrix (Equation 2-5). One such selection is



$\sum_J P_{ij}^2, \sum_J P_{ij}, N$

Figure 2-5. FAST Computation Overview

indicated by the symbol * in the inner loop environment distribution diagram of Figure 2-5. If the environment parameters are log-normally distributed, antilogs are taken at this point.

If azimuth sensitivity tables have been provided for any of the transfer functions, a single random number is used for selecting from all of the azimuth tables. Randomization of the transfer function for Component 2 is indicated in Figure 2-5.

When randomization on the environment parameters and transfer functions is complete, calculation of component probabilities of failure is accomplished as indicated by the nomographs of Figure 2-5. The fragility response is computed as follows:

$$R_i = TF_i \exp(e_i) \qquad (2\text{-}11)$$

The response parameter , R_i, is input to the fragility curve to obtain the component probability of failure.

Next the component probabilities of failure are combined, using the logic network compiler to obtain subsystem and system probabilities of failure.

The inner loop process is repeated until sample size or convergence criteria (discussed in the following section) are satisfied. The mean probability of failure is computed from the accumulated inner loop sample statistics for every component, subsystem and system. These sample means constitute the output from a single outer loop iteration. The process is repeated for many outer loop iterations, accumulating the probability of failure data (inner loop sample means) in histogram format. This process continues until outer loop sample size criteria (discussed below) are satisfied. The output from the outer loop iterations forms the survivability statistics.

2.2.5.2 Convergence Criteria

The data generated by the inner and outer loops can be represented by an array as follows.

$$
\begin{array}{lc}
 & \underline{\text{Row Averages}} \\
p_{11}\ p_{12}\cdots\ p_{1m_1} & \bar{p}_1 \\
p_{21}\ p_{22}\cdots\ p_{2m_2} & \bar{p}_2 \\
\vdots\quad\vdots & \\
p_{n1}\ p_{n2}\cdots\ p_{nm_n} & \bar{p}_n
\end{array}
\qquad (2\text{-}12)
$$

where the p_{ij} are failure probabilities. Each row in this array contains the probability of failure data for a system obtained from repeated iterations of the inner loop for one iteration of the outer loop. Each row is averaged to suppress the random variation within the row, yielding the $\bar{p}$ vector shown at the right above, whose elements are a sample from the distribution of system probability of failure perturbed by bias variations on the inputs.

The FAST code contains an algorithm to control the sampling accuracy from the random and bias distributions. The objective is to minimize computer cost without sacrificing necessary accuracy.

If too few iterations are done on the inner loop, the variance of the average probability of failure will be excessive and appear in the bias distribution of the outer loop, which is an undesirable condition. To control this, data on variability of system probability of failure within the inner loop is collected over the first 50 outer loop iterations to establish the number of inner loop iterations required to give the

specified accuracy. The inner loop convergence algorithm makes use of the following equation,

$$N_T(k) = \frac{\sum_{i=1}^{k} \left[\sum_{j=1}^{N_i} p_{ij}^2 - \left(\sum_{j=1}^{N_i} p_{ij} \right)^2 \frac{1}{N_i} \right]}{\epsilon_i^2 \sum_{i=1}^{k} (N_i - 1)} \qquad (2\text{-}13)$$

where

$N_T(k)$ = Estimated number of inner loop iterations required after k (≤50) outer loop iterations (Note: $N_T(k) \geq 5$ and $N_T(k) \leq 2N_T(k-1)$ are required).

N_i = Number of inner loop iterations that were done in the $i^{th} \leq k$ outer loop iteration

k = Number of outer loop iterations performed (k ≤ 50).

p_{ij} = Probability of failure from the j^{th} inner loop iteration of the i^{th} outer loop iteration

ϵ_i^2 = Maximum error bound on the inner loop average probability of failure

Equation (2-13) is applied repeatedly during the first 50 outer loop iterations. After 50 outer loop iterations, the number of inner loop iterations is fixed at the final value of $N_T(50)$.

Convergence of the outer loop is not tested until fifty iterations have been performed. At this time, the following computation is performed,

$$N(k) = \frac{\sum_{i=1}^{k} \bar{p}_i^2 - \left(\sum_{i=1}^{k} \bar{p}_i \right)^2 \frac{1}{k}}{\epsilon_{01}^2 (k-1)} \qquad (2\text{-}14)$$

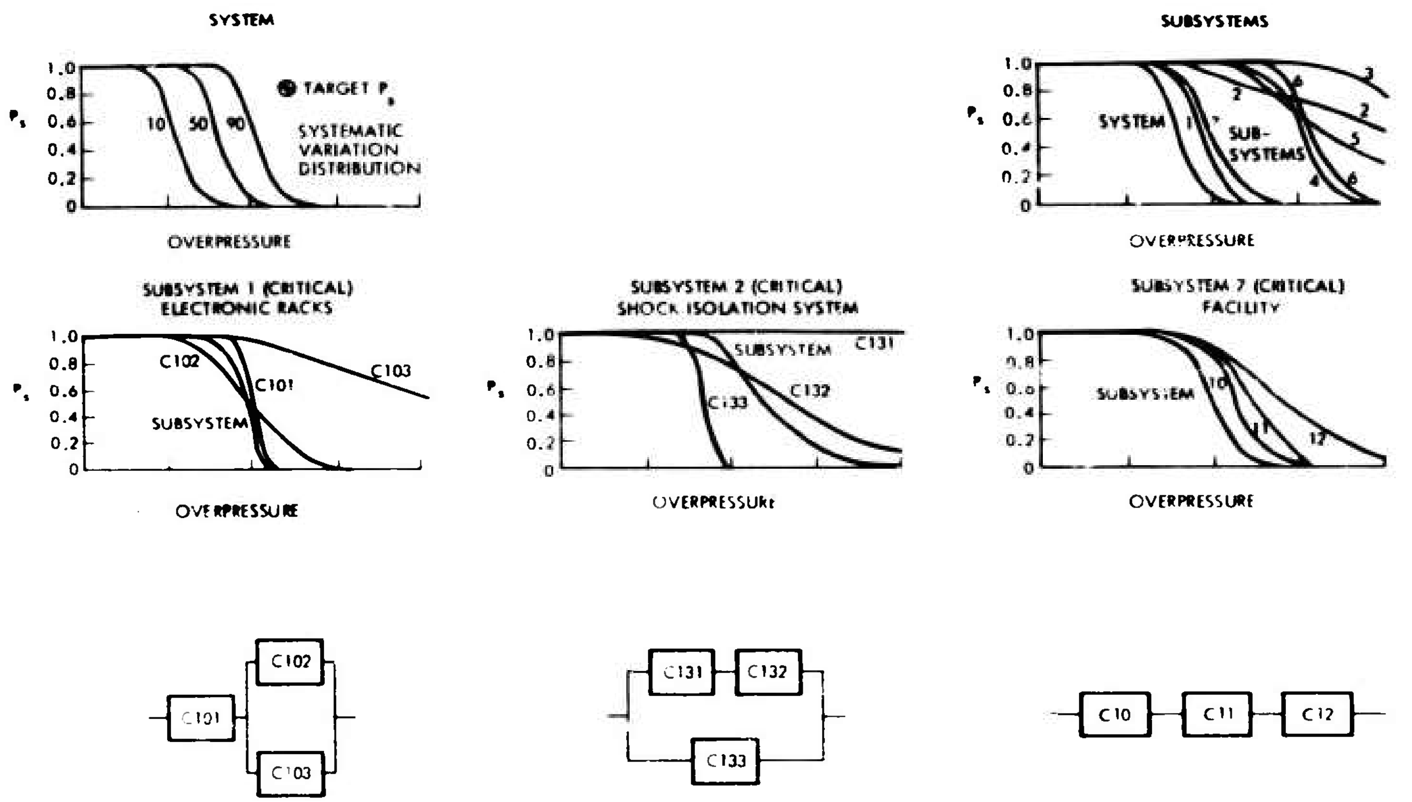


Figure 2-6. Typical Survivability Statistics

where

$N(k)$ = Estimated required number outer loop iterations

k = Number of outer loop iterations performed

ϵ_{01}^2 = Required error bound on the variance of the outer loop average probability of failure.

$\bar{p}_i$ = Mean of the i^{th} row in Eq. (2-12)

If N(k) is more than the number of outer loop iterations that have been performed, additional iterations are done in increments of 50 and the check is repeated until N(k) is less than the number of outer loop iterations performed.

After convergence on the mean is achieved, an additional test for convergence on the <u>median</u> is performed. This test uses the following

$$f(Ck + \sqrt{kC(1-C)}) - f(Ck - \sqrt{kC(1-C)}) < 2\epsilon_{02} \qquad (2\text{-}15)$$

where

k = Number of outer loops performed

C = 0.5, for the median

$f(x)$ = Fraction of the elements in the $\bar{p}$ vector of array (2-12) which are less then x $(1 \le x \le k)$

ϵ_{02} = Required accuracy on the median probability of failure.

If the inequality is satisfied, outer loop iteration stops.

The interpretation of (2-15) is as follows:

Given that k outer loops were performed, the $\bar{p}$ vector of array (2-12) is arranged in order of increasing magnitude. The resultant reordered array is called f(x), where x is the new index (low index, low magnitude).

Then equation (2-15) says that the average quantitative difference between two values of f, one on each side of the median, and the computed median is a measure of the uncertainty in the computed median. Additional outer loops are performed, in increments of 50, until the inequality (2-15) is satisfied or until a maximum of 625 outer loops have been performed.

The number of outer loop iterations required by the last criterion can become large as the average of the outer loop failure probability distribution approaches $C = 0.5$. This is the reason that the upper limit of 625 outer loop iterations has been imposed. This upper limit can be changed if a different value is appropriate for a given application.

2.2.5.3 Environment Parameter Vector Generation

The FAST code requires large numbers of vectors of environment parameters drawn from the distributions of systematic and random variations. The distributions are multivariate normal (after taking logarithms of log-normal environment parameters) with mean μ and covariance Σ. To provide computational efficiency and simplicity of programming, both the systematic and random covariance matrices are operated upon by a matrix decomposition approach. It is noted that the approach used in FAST requires that the matrix Σ be positive semi-definite (most routines of this type require Σ to be positive definite). This means that the code can accommodate analysis formulations wherein some environments are linear combinations of others. This section summarizes the theory of the matrix decomposition formulation.

A matrix Σ (which must be positive semi-definite) can be decomposed into a matrix C such that

$$C \tilde{C} = \Sigma \tag{2-16}$$

where $\tilde{C}$ is the transpose of C. Then the required random vector x is calculated from a vector y (whose elements are independently and normally distributed random variables with zero means and unit variances) as follows:

$$x = \mu + Cy \tag{2-17}$$

To see that x has the desired distribution, assume $C\tilde{C} = \Sigma$ and let $x' = Cy$, where y is a vector of uncorrelated random variables with mean zero and unit variance (and hence, $E\{y\tilde{y}\} = I$, where $E\{\cdot\}$ denotes expectation). Then

$$\begin{aligned} E\{x'\tilde{x}'\} &= E\{Cy\tilde{y}\tilde{C}\} \\ &= C\,E\{y\tilde{y}\}\tilde{C} \\ &= C\,I\,\tilde{C} \\ &= C\tilde{C} \\ &= \Sigma \end{aligned} \tag{2-18}$$

Thus, x' has zero mean and covariance Σ. Finally, setting $x = \mu + x'$ completes the demonstration.

The method for computing C from Σ, known as the square root method, is given in Reference (4), which should be consulted for the basic theory. Most of the algorithms in the literature require that Σ be positive definite. The matrix decomposition algorithm in this analysis incorporates a modification which relaxes that requirement, so that Σ can be positive semi-definite (i.e., its rank, r, can be less than its size, n). In the modified routine, each successive pivotal element, C (k,k), is selected as the largest remaining leading diagonal element rather than in numerical order as in the standard algorithm. Thus, if Σ has rank $r<n$ (the size of Σ) after r applications of the algorithm, the remaining leading diagonal elements, C (k,k), will be approximately zero (i.e., equal to zero except for computational errors due to rounding). Rows containing diagonal elements, $C(k,k) = 0$, are set equal to zero. After the decomposition routine has been completed, the matrix contains r non-zero diagonal elements and n-r zero diagonal elements. Then rank of C (and Σ) is r.

2.2.5.4 Output Displays

Results of FAST calculations can be displayed in a variety of fashions as illustrated in Figure 2-6. One important display is that of the system survivability together with associated uncertainty, as a function of overpressure or miss distance. (See upper left of Figure 2-6). In the upper right of this figure is plotted the median value of the probability of survival of the system and its subsystems, as functions of overpressure. This output format is valuable for identifying subsystems which are weak link items, and hence potential candidates for hardening.

The other three diagrams of the figure each show the median probability of survival of a critical subsystem, together with the medians of the components, all as functions of overpressure. This format enables identification of components which are weak link items in the subsystem. Taken together, these displays enable identification of those subsystems and components contributing most to the failure of the system.

2.3 SYSTEM MODELING CONSIDERATIONS

The derivation of inputs for probabilistic assessments of a system's capability to function after being subjected to hostile environments can proceed in several ways. No single approach is best for all systems, as some problems naturally emphasize environments, other problems highlight transfer functions or fragilities. It is necessary to take a broad view prior to developing the mathematical model of the system, since the approach which is taken in formulating the problem will influence the system network.

The first consideration is to specify the definition for system failure or, conversely, system survival. A system is generally said to survive if it retains the capability to perform its mission for a stated period of time after being subjected to one or more applications of hostile environments. Some systems may have multiple missions, such as both tactical and strategic, each of which utilizes different combinations of subsystems. Another definition of system survival (repairable survival) might allow a system to be inoperable for some stated time interval provided that it could be returned to service

(i. e., regain the capability to perform its mission) within the time interval. Obviously, the differences between single and multiple attack conditions and emergency, extended and repairable survival definitions will greatly affect the derivation of inputs for FAST code system simulations.

This section considers various techniques and practical considerations for developing the system network, fragilities, environments, and transfer functions. To do this, the following data needs to be considered:

a. Environments

- Median values for baseline threats.
- Random and systematic variation covariance matrices for baseline threats.
- Scaling parameters

b. Transfer functions relating environments with corresponding fragility parameters

- Best estimate values
- Uncertainty bounds

c. Fragility curves for each network element

- Best estimate curve
- Bounding curves

d. System network.

2.3.1 Environments

As previously discussed, the FAST code accepts a probabilistic definition of environments, with the multivariate log-normal distribution used to define the two kinds of variation treated: random and systematic variation. If the environments are considered to be defined by first order (mean) and second order (variance) moments of distributions which are better represented in log-space than in linear space, then the log-normal distribution is both physically plausible and convenient.

Environments may include both ambient and in situ conditions, as well as characteristics of the hostile environment. For example, geologic properties such as depth to water table may be treated as an environment. The water table depth could be a very significant parameter for components sensitive to facility flooding. Hostile environments such as peak values of free field displacements, velocities and accelerations, or shock spectra parameters when defined at appropriate depths, can be related to critical internal shock environments or system responses. Ground shock loads on protective structures may be related to free field stresses and so on. In general, environments are a mix of ambient and in situ conditions transformed into hostile environments by the energy associated with a catastrophic event.

Among the most important aspects of environment definition is the derivation of covariance matrices for random and systematic variations on environment predictions. Covariance or correlation is important because non-zero covariance between two environments can strongly affect the joint probability that both environments exceed some set values at the same time. For a series system where two components are affected by strongly correlated environments (e.g., debris and crater-related ground shock), then either both components fail or both survive, and the probability of system kill does not compound itself as more components are considered. Alternatively, if a series system is subjected to anticorrelated hostile environments (e.g., air induced velocity and crater-related velocity), then the probability of system kill can be substantially greater than if the environments were uncorrelated. The effects of covariance on parallel and series-parallel systems are also important.

Noticing the importance of covariance, how does one go about deriving covariance matrices for random and systematic variations on the environment predictions? The answer depends upon the problem solving mode.

For a well known system in-place at many sites, one can perform environment parameter predictions based upon known in situ properties and computer code calculation results and/or empirical scaling laws.

A vector of pertinent environments can be constructed for each site, and multivariate log-normal statistics can be computed (Equation (2-6)) for one or more groups of these sites to define the mean vector(s) and random covariance(s) of the environment predictions. Next, one examines the environment prediction equations (which are functions of <u>in situ</u> conditions and the manner of energy application) and determines the underlying systematic variation or uncertainty on material properties, loads and analysis (prediction) technique. If the prediction equations are linear (in log-space), the environment systematic variation covariance, Σ_{uu}, may be computed directly from the covariance of underlying variables, Σ_{vv}

$$\Sigma_{uu} = L \, \Sigma_{vv} \tilde{L} \tag{2-19}$$

where L is the matrix which defines the linear prediction technique in log-space.

For example, consider a system sensitive to only two environments; vertical velocity (v_z) and horizontal velocity (v_h) defined by

$$v_z = \frac{P}{\rho_m c}, \quad v_h = \frac{P}{\rho_m U}$$

where P is the airblast pressure, U is the air shock speed, ρ is the media density and c is the media stress wave speed. In log-space the prediction equations are

$$\log v_z = \log P - \log \rho_m - \log c$$

$$\log v_h = \log P - \log \rho_m - \log U$$

Thus,

$$L = \begin{bmatrix} 1 & -1 & -1 & 0 \\ 1 & -1 & 0 & -1 \end{bmatrix}$$

and

$$\Sigma_{vv} = \begin{bmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \rho_{13}\sigma_1\sigma_3 & \rho_{14}\sigma_1\sigma_4 \\ & \sigma_2^2 & \rho_{23}\sigma_2\sigma_3 & \rho_{24}\sigma_2\sigma_4 \\ & & \sigma_3^2 & \rho_{34}\sigma_3\sigma_4 \\ \text{SYMMETRIC} & & & \sigma_4^2 \end{bmatrix}$$

where the subscripts 1, 2, 3, 4, refer to log P, log ρ_m, log c and log U, respectively.

An alternate mode of generating random covariance matrices for environments is to estimate the random variation of underlying variables, and propagate these random covariances through the prediction equations as was described for systematic variations.

Scaling laws for nuclear weapons effects have been included in FAST, in the forms specified in Equations (2-7a and 2-7b). For surface bursts, these forms of scaling are adequate for reasonable intervals of range, R (and corresponding overpressure, P); yield, W; and height of burst, H. The forms of scaling have the property that logarithmic covariances are constant while logarithmic means vary linearly as the logs of overpressure, range, or yield. More complex forms of scaling both the mean and the covariance are possible. However, only simple scaling laws have been implemented in the FAST code because they have proved to be adequate in all past applications. The effects of complicated scaling can be modeled by computing as many baseline threat mean vectors and covariance matrices as required, or, should some new application dictate, the scaling can be changed as it has been for nonlinear overpressure scaling.

Scaling for height of burst is accomplished by scaling pressure and related effects with overpressure and yield separately for each H, and by scaling crater-related effects as a function of H, based on the crater volume scaling V = V (H).

2.3.2 Transfer Functions

Transfer functions normally take a single free field environment and convert it into a critical local response. To illustrate a more complex case, consider a buried structure which is subjected to vertical overpressure, P, on the roof and horizontal pressure, $K_o P$, on the walls. The lateral pressure coefficient K_o is a property of the <u>in situ</u> soils, and as such proper correlation of environments requires that K_o be treated as an environment although it obviously would not scale with pressure, yield, or HOB. Now the fragility can be expressed as a function of the dimensionless static ratio of stress to allowable stress, S,

$$S_r = \frac{\tau_r}{V_r} = \frac{\lambda_r P}{V_r} \text{ for the roof} \tag{2-20a}$$

and

$$S_w = \frac{\tau_w}{V_w} = \frac{\lambda_w K_o P}{V_w} \text{ for the wall} \tag{2-20b}$$

where λ_r and λ_w denote the static stress concentration factors for the roof and the wall and V_r and V_w denote the allowable shear strengths of the concrete roof slab and the concrete wall. Based on construction data, one might find or anticipate that V_w and V_r are correlated due to workmanship, weather, or other factors. In order to preserve this correlation one could define P/V_r and PK_o/V_w as environments.

Carrying the correlations even further, the stress concentration factors λ_r and λ_w might have been derived from the same set of data on shear failures of deep slabs. In this case, an error in data interpretation would manifest itself in both λ's, hence these correlations might be best preserved by calling $\lambda_r P/V_r$ and $\lambda_w K_o P/V_w$ environments. Then the transfer function defines only the dynamic amplification factors for the roof and wall, respectively, and the fragility curves might be very simple curves centered on the dimensionless constant 1.

It is only the ratios of response to allowables which matter. Statistics on these ratios will be reasonable if the steps which are used in developing environments, transfer functions, and fragilities are reasonable. Transfer function effects should not be arbitrarily lumped into either environments or fragilities. Nevertheless, the distinctions between environments, transfer functions, and fragilities are sometimes left to the convenience of the user, and one should not let semantics limit one's flexibility in deriving inputs for the FAST code.

2.3.3 Fragilities

Derivation of fragilities starts with the frequently difficult task of identifying the critical response for each component. Fragility curves are then determined, using analysis, test data or a combination of analysis and test. These fragility curves represent the best estimate of the actual failure probability. Each fragility must be carefully developed to ensure that it is not too low or too high. In general, engineers and analysts tend to produce fragilities which are conservative (too low). Special precautions are strongly recommended to provide fragilities which are realistic. Specific examples provide the best guide to understanding fragility derivation, and two examples are highlighted in the sample problem of Section 6.4.

In FAST there is considerable flexibility in the representation of fragilities. This flexibility facilitates specialization where component or subsystem fragilities are derived by experts independent of the derivation of other fragilities, transfer functions, and environments. However, trouble can ensue if a transfer function is misidentified as a fragility. When the problem is properly subdivided and environments and transfer functions are properly derived, the development of fragilities is facilitated.

Azimuth sensitivities can be introduced in series chainwise fashion, which enables use of multiple azimuth tables to model complex situations. One source of significant azimuthal effects are drag loads on blast doors or non-surface flush and non-axisymmetric structure conditions. Non-flush surfaces may contain multiple failure modes (i.e., blast, thermal, debris, shock, radiation) and all these modes usually occur concurrently.

2.3.4 Network

Development of the system network involves determination of the series/parallel relationship among component fragilities. In addition, the manner in which components are grouped to form subsystems needs to be established. A few of the possible approaches to organizing a study are to group into logical subsystems

a. components which perform related functional relationships,

b. components which are affected by related environments,

c. components which are in nearby locations, and

d. components which have similar attack-failure-repair chronology.

The manner in which components are grouped can be quite important since the survivability statistics on subsystems, as well as on components and on the system, are output from FAST. By the use of ganging it is possible to run a problem in which one system is organized from several logical viewpoints. Often this combined approach will maximize the user's understanding of the survivability statistics.

In order to model component fragility over a wide range of conditions, it is often necessary to expand the system network. For example, a component may be sensitive to displacements for high weapon yield conditions and velocities for low weapon yield conditions. Such a component would be best modeled by two fragilities in a series or parallel relationship (depending on the specifics of the example). Impacts between objects and combinations of stress and deformation also can result in complicated logic. Sometimes network logic complications can be removed or reduced by using more complex transfer functions and/or more complex fragilities. The requirements of the study and characteristics of the system should play an important role in deciding how the modeling is performed.

Other changes from the functional system network to the FAST system network are dictated more directly by the questions which the FAST calculations are expected to help answer. For example, one

might want to find out how often Component A and Component B both fail even though the failure of either would result in system failure. To calculate joint failure probabilities, one could construct a dummy system (using the ganging technique) containing Components A and B in parallel.

In finalizing a network, one should prune a preliminary fault tree, making use of quantitative knowledge of fragilities and environments to eliminate superhard components and components which almost always fail in conjunction with other more critical components. In this manner complex systems are meaningfully reduced to manageable proportions. Complete identification and documentation of even superhard items should be accomplished as part of the network synthesis prior to pruning the fault tree.

3.0 INPUT DESCRIPTION

This section defines all of the inputs required for the FAST Code and describes the required input format.

The inputs are contained on five classes of cards

- Fragility - Transfer Function Cards
- System Network Cards
- Environment Parameter Cards
- Threat Condition and Convergence Criteria Cards
- Miscellaneous and Control Cards

Each kind of input card will be discussed, giving the card format and a table defining all variables. Each format is illustrated by cards taken from the Sample Problem baseline run in Appendix A, unless otherwise indicated.

In discussing the inputs, any number that is not identified as an integer will be a real (floating point) number. All integers should be written without a decimal point or exponent. Real numbers may be written in any of the equivalent forms.

123 = 123. = 123.0 = 1.23E2

or -3.2E-1 = -0.32

3.1 FRAGILITY - TRANSFER FUNCTION CARDS

The basic parameters that can be defined on the fragility - transfer function card are summarized in Table 3-A. For convenience, this card is referred to as a component card. The basic form of the component card is

$$\text{Ci} = /0,\ f_{p_{L,1}}, f_{p_1}, f_{p_{U,1}}\ /pf_2,\ f_{p_{L,2}}, f_{p2}, f_{pU,2}/$$

$$\ldots /1,\ f_{p_{L,6}}, f_{p_6}, f_{p_{U,6}}\ /RK = ne_i,\ tf_i,\ tfu_i\ \$$$

The information from a typical component card is reproduced below with a description of each parameter on the card.

C101 = /0, 30, 35, 40/1, 40, 45, 50/R1 = 13, .2, 1.2 $

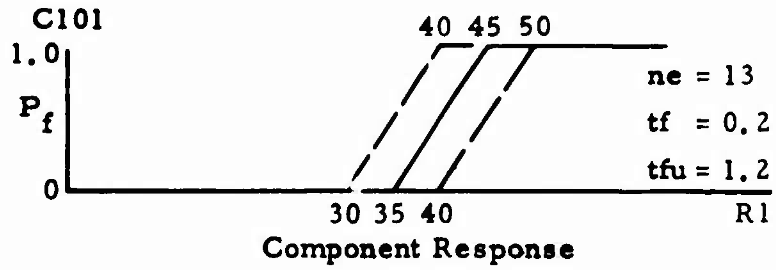
where C101 = is the component card identification. Components are always identified by integer numbers.

Table 3-A. Fragility - Transfer Function Parameters

Parameter	Description
i	Integer designating component number
pf_j	The probability of failure at the j^{th} level
$f_{p_{L,j}}$, f_{p_j}, $f_{p_{U,j}}$	Piecewise linear representation of the fragility. $f_{p_{L,j}}$, f_{p_j}, $f_{p_{U,j}}$ are the corresponding lower bound, nominal and upper bound values of the component response, respectively, at pf_j. A maximum of 6 points can be considered. $f_{p_{L,j}}$, and $f_{p_{U,j}}$ need not be specified if there is no systematic variation
ne_i	Environment parameter reference number corresponding to component i
tf_i*	Transfer function
tfu_i*	Transfer function k factor
k	Storage register for transformed environment
$	Terminates component card and may be followed by comments

*Need not be specified if equal to unity

/0, 30, 35, 40/1, 40, 45, 50/ represents the following fragility curve:



R1 = 13, .2, 1.2 describes the relationship between an environment and a component response. Environments, like components, are referenced by integer numbers. In this case, the reference is to environment number 13. Of E_{13} is the value associated with environment 13, then the transfer function, tf = 0.2, is used as follows and the result temporarily stored in R1

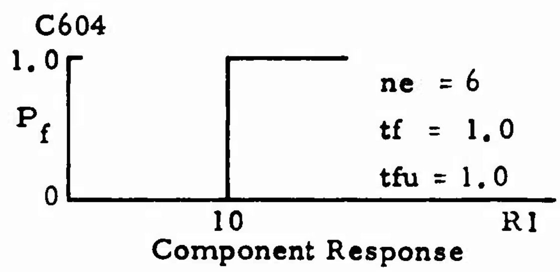
$$R1 = E_{13} * 0.2$$

The transfer function k factor, tfu = 1.2, applies bias to R1. The result is then stored back into R1.

The information on a component card could be as simple as

C604 = /0, 10/1, 10/R1 = 6, 1, 1 $

which represents a vertical (cookie cutter) best estimate fragility. The component response is equal to the unmodified environment 6, since the tf and the tfu are both equal to unity. It is important to note that all fragilities must be defined at $P_f = 0.0$ and $P_f = 1.0$. The fragility curve for this component is



The tf and tfu values need not be punched when they are equal to 1, so the following forms are also equivalent for C604:

C604 = /0, 10/1, 10/R1 = 6 $
C604 = /0, 10/1, 10/R1 = 6, 1 $
C604 = /0, 10/1, 10/R1 = 6, , 1 $

Fragility curves and transfer functions that are more complex can be handled in FAST. These are discussed in the following section. This section can be skipped during initial acquaintance with the code.

3.1.1 Special Functional Operations

The environment field can have more complex expressions including the use of CAN and USER. CAN provides the option to add environments or the transformed environments and USER provides azimuth sensitivity. The first example illustrates the use of CAN (Appendix A does not include an example):

C502 = /0, 2, 5, 7/1, 3, 6, 9/R1 = 3/R2 = 6/R3 = 2, 0.2/CAN(3)/ R1 = R1, 1, 1.2 $

where R1 = 3 stores environment 3 in R1

R2 = 6 stores environment 6 in R2

R3 = 2., .2 stores 0.2 times environment 2 in R3

CAN(3) indicates how many environments to add and stores the result in R1. Thus, R1 = R1 + R2 + R3.

R1 = R1, 1, 1.2 applies the transfer function k factor to R1 and stores result back into R1.

The input USER(IØP) supplies the calling sequence parameter IØP to a subroutine called USER which must be provided by the program user in order to account for azimuth sensitivity. The parameter IØP then allows for azimuth table selection. An example of such a subroutine is given in Appendix A.

Component card C102 illustrates a reference to subroutine USER.

C102 = /0, 9, 10, 11/1, 11, 12, 13/R1 = 17/USER(1) = R1, 1.0, 1.2$

where R1 = 17 indicates environment 17 is stored in R1

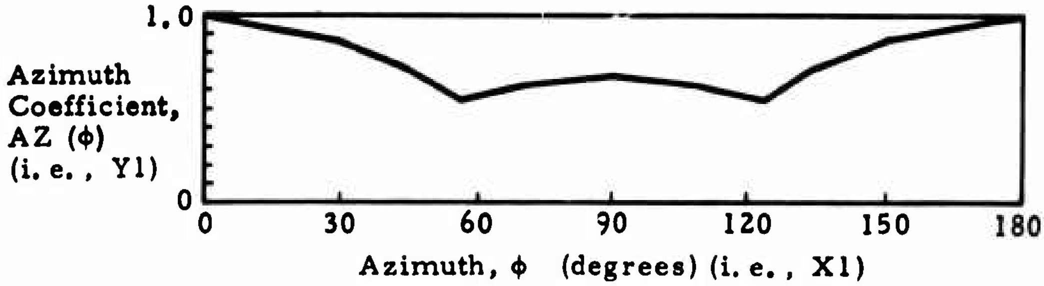
USER(1) indicates the azimuth table to apply for ne_{17} in C102 (see the following for example).

Azimuth table taken from subroutine USER (see Appendix A):

X1 /0, 30, 45, 56, 60, 70, 90, 110, 120, 124, 135, 150, 180, 210, 225, 236, 240, 250, 270, 290, 300, 304, 315, 330, 360/

Y1 /1, .866, .707, .552. .578, .627. .667, .627, .578, .552, .707, .866, 1, .866, .707, .552, .578, .627, .667, .627, .578, .552, .707, .866, 1./

The previous table represents the following:

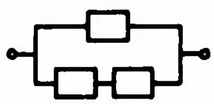
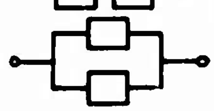
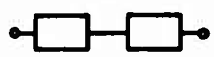


To be consistent with CAN, USER stores the result in R1
R1 = R1, 1.0, 1.2 indicates the k factor is applied and the result stored in R1.

3.2 SYSTEM NETWORK CARDS

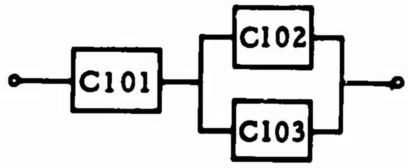
Table 3-B summarizes the Boolean network parameters.

Table 3-B. Network Parameters

Parameters	Example	Description
+		Boolean operator for combining series elements
*		Boolean operator for combining parallel elements
( )		Used to order how elements are combined
Ci or i	C101 or 101	An integer that identifies a component element. The C need not be punched
SSj	SS 1	An integer than identifies a sub-system element
SYSk	SYS 100	An integer that identifies a system element. The results, when printed, are identified by negative k (-k)
$		Terminates the equation

The first two examples are selected from Appendix A.

SS1 = 101 + 102 * 103 $

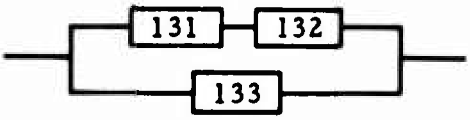


The operator * takes precedence over +, thus the probabilities of failure $P_{102} * P_{103}$ are combined first to compute a combined probability of failure, P_r. Then P_r is combined with P_{101} as follows:

$$P_{SS1} = 1 - (1-P_{101})\,(1-P_r)$$

Parentheses are used to change the order of operations:

SS2 = (131 + 132) * 133 $



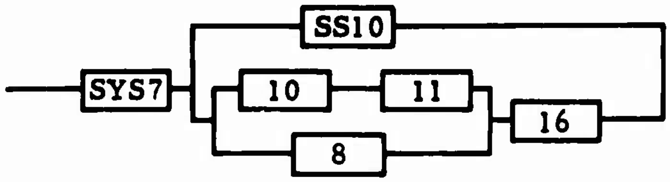
In this case P_{131} and P_{132} are combined first to compute a combined probability of failure: $P_r = 1 - (1-P_{131})(1-P_{132})$. Then P_r is combined with P_{133} as follows:

$$P_{SS2} = P_r * P_{133}$$

In each of these examples the result is designated as a subsystem (SS1 and SS2), but they could have been systems (SYS1 and SYS2). The following examples are not typical, but they illustrate that parentheses can be nested (to almost any depth) and any elements can be combined as shown by the following equations:

SS6 = SYS7 + SS10*((10+11)*8 + 16) $

SS6 = SYS7 + SS10*((10+ 11)*8 +16) $



SYS10 = SYS8 + SS6 + 21 $

There are two network restrictions. First, just as components must be entered before they are used in an equation, subsystems and systems must be defined (equations must be in logical order as in a FORTRAN program). Second, redundant parentheses are not permitted:

(101) $

((101 + 102))*103 $

(101 + 102 + 103) $

3.3 ENVIRONMENT PARAMETER CARDS

Environments are entered on various cards. Each environment parameter is described in Table 3-C.

Each of the vectors in Table 3-C can define up to 35 environments. The only card that is not illustrated in Appendix A is the ENV card. This card is an alternate method of entering and/or changing environment parameters; the general format is:

ENV i = MODE(i), ULN(i), K-FAC(i), ALPHA(i), BETA(i), GAMMA(i), HB(i), GAMMAB(i)

(The last three parameters need not be entered if they are not used; just as GAMMAB and HB-GAMMAB are not required.)

To change originally punched and entered environment parameters, the following cards, which supersede the previous information, are entered after the original cards

ENV2 = 1, .5, 2, .1, .05, .8, .02, .1

ENV3 = 2, .6, 1.5, .2, .1, .7, .2, .7

Table 3-C. Environment Parameters

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>MODE</td><td>Vector which identifies, by integer numbers, the type of environment and scaling option
<table>
<tr><th>MODE</th><th>Scaling Option</th><th>Type of Environment</th></tr>
<tr><td>1</td><td>range</td><td>log-normal</td></tr>
<tr><td>2</td><td>range</td><td>normal</td></tr>
<tr><td>3</td><td>range</td><td>normal (log base 10)</td></tr>
<tr><td>4</td><td>overpressure</td><td>log-normal</td></tr>
<tr><td>5</td><td>overpressure</td><td>normal</td></tr>
<tr><td>6</td><td>overpressure</td><td>normal (log base 10)</td></tr>
</table>
</td></tr>
<tr><td>ULN</td><td>Vector of logarithmic environment medians</td></tr>
<tr><td>ALPHA</td><td>Vector of yield scaling exponents</td></tr>
<tr><td>BETA</td><td>Vector of range or pressure scaling exponents</td></tr>
<tr><td>GAMMA</td><td>Vector of coefficients to scale height of burst ($HOB/W^{1/3}$)</td></tr>
<tr><td>HB-GAMMAB</td><td>Height of burst break point and scaling coefficient are entered in pairs into a vector</td></tr>
<tr><td>S</td><td>Used to enter rows or columns of symmetric random variation covariance matrix</td></tr>
<tr><td>K-FAC</td><td>Vector of K factors for systematic environment variation</td></tr>
<tr><td>A</td><td>Used to enter rows or columns of symmetric correlation matrix</td></tr>
<tr><td>ENV</td><td>Alternate parameter for entering several environment parameters</td></tr>
</table>

or the equivalent alteration cards are entered after the original cards

MODE(2) = 1, 2, ULN(2) = .5, .6, K-FAC(2) = 2, 1.5
ALPHA(2) = .1, .2, BETA(2) = .05, .1, GAMMA(2) = .8, .7
HB-GAMMAB(2) = .02, .1, .2, .7

As Appendix A shows, the formats for the environment parameters are not complex. Thus, the MODE card will be discussed to represent the vector inputs and the S card will be discussed to represent the matrix inputs.

MODE = 16 * 4, 3 * 1

This is a convenient format that is equivalent to:

MODE = 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 1, 1, 1,

The integer number following the S identifies the diagonal element of the matrix. For example:

S17 - .0230, 0.107, .0180

which defines the following elements

$$s_{17,17} = .0230$$
$$s_{18,17} = 0.107$$
$$s_{19,17} = .0180$$

3.4 THREAT CONDITIONS AND CONVERGENCE CRITERIA CARDS

Table 3-D defines the threat condition parameters.

The choice for the sample problem was to enter the following cards to describe the threat.

NOMPWH = 600, 5 $

where $P_o = 600$, $W_o = 5$ and $H_o = 0$

and W = 5

OP = 200, 400, 500, 600, 700, 800, 1000, 1200

Table 3-D. Threat Condition Parameters

Parameter	Description
NOMRWH	Name used to identify the input parameters R_o, W_o and H_o. NOMRWH = R_o, W_o, H_o
NOMPWH	Name used to identify the input parameters P_o, W_o and H_o. NOMPWH = P_o W_o, H_o. Only one of these two cards (NOMRWH or NOMPWH) is required
R_o	Nominal range
W_o	Nominal yield
H_o	Nominal height of burst - assumed to be zero if not entered
P_o	Nominal overpressure
R	Ranges of interest (see OP)
W	Yield of interest
H	Height of burst of interest - assumed to be zero if not entered
OP	Vector of overpressures of interest. Either R or OP is given, not both

for yield of 5 MT and various pressures (ranges are calculated automatically by the program as is R_o).

Table 3-E defines the convergence criteria parameters.

For the sample problem, the choice was to use the default values. The default values are the values automatically used by FAST if no other specific values are included in the input. The default values, in the same order presented in Table 3-E, are the same as entering the following card.

CONFI = 5, 50, 50, .05, .03, .02, .5

Table 3-E. Convergence Criteria Parameters

Parameter	Description
CONFI	Name of the card which provides the parameters CONFI=NIMIN, NIMAX, NOMIN, EPSI, EPSO, EPSOC, FRACTILE
NIMIN	Number of inner loop iterations to do before testing inner loop convergence (integer number)
NIMAX	Number of outer loop iterations to do before accepting the inner loop convergence (integer number)
NOMIN	Number of outer loop iterations before testing outer loop convergence (integer number)
EPSI	Required standard deviation for inner loop convergence test
EPSO	Required standard deviation for convergence of outer loop mean (first outer loop test)
EPSOC	Required standard deviation for convergence of median (0.5) or more fractile (second outer loop test)
FRACTILE	Vector of fractiles from 1 through 7

3.5 CONTROL CARDS AND MISCELLANEOUS CARDS

The control cards cause the program to perform certain operations, which are completely described in Table 3-F.

Table 3-F. Control Cards

Card	Description
RUN$	Indicates that all the information for a case has been entered.
RESET$	Indicates the beginning of all new information — previous information is deleted.
STOP$	Indicates the end of data and causes the program to stop.
PLOTON=s$	Turns the printer plot option on, where s is the overpressure per inch to scale the plot abscissa, which is six inches long.
PLOT$	Indicates when to plot the results from one or more cases.
PLOTOFF$	Turns the plot option off.

Table 3-G defines some miscellaneous cards and features. Some of these cards are not commonly used but are available.

Table 3-G. Miscellaneous Cards and Features

Card/ Feature	Description
RANUN	RANUN=NUMUR, NUMNR. On some computer systems the starting values for the random number generators (uniform, and normal) can be changed, but the user must be familiar with FAST and the available generators. The default values assumed by the TRW random number generators are 1.
RECODE	This card allows the data to be punched on both an 029 and 026 keypunch machine
Ci=$	Deletes Component i
SSj=$	Deletes Subsystem equation j
SYSh=$	Deletes System equation h
null-field	A null-field is described below
L format	For MODE=3 or 6 the ULN fragility and transfer function will be in log base 10. If the L format is used in place of the E, the program will take $\log_{10}$. 18L0 = 1.8 L1 = 180L-1 where the result is: $1.8L1 = \log_{10} (1.8*10^1) = 1.25527$

The RECODE card is fixed format (no blanks) that starts in column 1.

RECODE=ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789,+-x/.()=$

If the data deck has been punched on both an 029 and 026 keypunch, then the RECODE card will be useful. In front of 029 cards, place a RECODE card punched in 029 and an 026 RECODE card in front of 026 cards.

A null-field is defined as a numeric field that does not contain a number. This is useful for changing values on cards like CONFI or ENV. As an example, this card will only change the values for NOMIN (3rd) and EPSO (5th)

CONFI=2*, 40,, .02

3.6 CASE STACKING

The FAST inputs can be manipulated to do ganging, job stacking and case stacking. Ganging permits several system configurations in one case. The RESET$ card deletes previous information, which permits stacking jobs in a computer run.

Appendix A (yield and HOB perturbations) shows a typical example of case stacking, where the yield and height of burst are varied. In other words the information in a previous case is available for the next case and can be modified as required. For example, components and network equations can be replaced or deleted. Thus C604, from Appendix A, (Baseline System Sample Problem) could be changed to

C604=/0, 10/1, 10/R1 = 6, 1.2$

or deleted (be sure to modify network) as shown below:

C604=$

4.0 OUTPUT DESCRIPTION

All of the input data is printed back out. Since the "USER(IOP)" subroutine is an integral part of the input, the subroutine is printed out to show the azimuth data; input cards are printed to show information on the component, network, and environment cards.

If there are any keypunch errors or errors of omission or inconsistencies, error messages are printed out. See Section 4.2 for a list of the messages and suggestions for correction. Finally, the actual calculated data for the probability of survival for each component, subsystem, and systems are displayed.

4.1 SURVIVABILITY RESULTS

If the rank for either the random (S) or systematic (A) variation matrix is less than the dimension, then the following illustrates what will be printed.

S MATRIX - DIMENSION 12 HAS RANK 11

For each range and overpressure, the input ULN vector is scaled and the median values of the environment are displayed as follows:

ULN(0)=	8.00000E+02
ULN(1)=	1.96435E+01
ULN(2)=	5.06580E+02
ULN(4)=	7.60276E+02
ULN(5)=	1.53933E+01
ULN(6)=	5.09239E+00
ULN(8)=	7.73887E+01
ULN(10)=	4.02720E+03
ULN(11)=	2.07585E+03
ULN(12)=	1.33311E+02
ULN(13)=	2.66582E+02
ULN(14)=	8.00056E+02
ULN(15)=	8.00056E+02
ULN(6)=	8.00056E+02
ULN(17)=	2.18480E+01
ULN(18)=	3.56735E+01
ULN(19)=	1.79089E+01

Note that ULN (0) is not an input but is the current value for overpressure, which can be referenced on the component cards as environment 0, i.e. R1 = 0. The information on the condition of the threat at which the above environments are calculated are displayed as follows:

W= 5.0000E+00 H= 0. R= 5.1909E-01 P= 6.0000E+02

where

W = weapon yield, MT
H = height of burst, K-ft
R = miss distance, NM
P = overpressure, psi

During the Monte Carlo process, the code tests for convergence as described in Section 2.2.5.1. If there is no convergence, a line of information is printed which includes the number of inner loop iterations (NI), number of outer loop iterations (NO) and the test criteria (EPS). EPS is ϵ^2 (equation 2-13), ϵ_{01}^2 (equation 2-14) or $2\epsilon_{02}$ (equation 2-15). If the process converges within 625 outer loop iterations, then a line is printed to display the number of iterations. A sample appears below. This process can be controlled by the CONFI input card (see Section 3.4).

```
W= 5.0000E+00 H= 0.          R= 4.6795E-01 P= 8.0000E+02
NO CONVERGENCE - NI=      5 NO=        0 EPS= 5.0E-02
NO CONVERGENCE - NI=      5 NO=        1 EPS= 5.0E-02
NO CONVERGENCE - NI=      5 NO=        2 EPS= 5.0E-02
NO CONVERGENCE - NI=      5 NO=        3 EPS= 5.0E-02
NO CONVERGENCE - NI=      5 NO=        4 EPS= 5.0E-02
NO CONVERGENCE - NI=      5 NO=        5 EPS= 5.0E-02
NO CONVERGENCE - NI=      5 NO=        6 EPS= 5.0E-02
NO CONVERGENCE - NI=      5 NO=        7 EPS= 5.0E-02
      .              .         .               .
      .              .         .               .
      .              .         .               .
NO CONVERGENCE - NI=      5 NO=       41 EPS= 5.0E-02
NO CONVERGENCE - NI=      5 NO=       42 EPS= 5.0E-02
NO CONVERGENCE - NI=      5 NO=       43 EPS= 5.0E-02
NO CONVERGENCE - NI=      5 NO=       44 EPS= 5.0E-02
NO CONVERGENCE - NI=      5 NO=       45 EPS= 5.0E-02
NO CONVERGENCE - NI=      5 NO=       46 EPS= 5.0E-02
NO CONVERGENCE - NI=      5 NO=       47 EPS= 5.0E-02
NO CONVERGENCE - NI=      5 NO=       48 EPS= 5.0E-02
NO CONVERGENCE - NI=      5 NO=       49 EPS= 5.0E-02
   CONVERGENCE - NI=      5 NO=       50
```

The next FAST display is the histogram shown in Table 4-A. The column to the far right lists the input numbers which identify components, systems (negative numbers) and subsystems. There are twenty-two columns to store probability numbers for the systematic variation distribution. The first column is for the count of the probability of survival (P_s) equal to 0 and the last column is for $P_s = 1.0$, while the twenty divisions between them are incremented by steps of 0.05 probability. The number of outside loops determine the total number of marks that are put into all increments. The average probability of the inside loop determines which increment gets the mark for each outside loop iteration.

The following printout section gives the cumulative distribution of systematic variation by individual components, subsystems and systems, and are ranked by the 50 percentile or median of the distribution. The component, subsystem and system names (numbers) are listed under NAME on the left hand side with columns designated 1, 10, 30, 50, 70, 90, 99 for the percentile of the distribution for each component, etc. An interpolation routine is used which prints this information by normalizing to the number of iterations actually performed.

INDIVIDUAL COMPONENTS

NAME	1	10	30	50	70	90	99
133	0.000	0.000	0.000	0.000	0.000	0.000	1.000
101	0.000	0.000	0.000	0.000	0.000	.200	1.000
706	1.000	1.000	1.000	1.000	1.000	1.000	1.000
9	.825	1.000	1.000	1.000	1.000	1.000	1.000
705	0.000	0.000	.850	1.000	1.000	1.000	1.000

INDIVIDUAL SUBSYSTEMS

NAME	1	10	30	50	70	90	99
1	0.000	0.000	0.000	0.000	.400	1.000	1.000
8	0.000	0.000	0.000	.100	.475	.962	1.000
7	.002	.017	.050	.167	.340	.583	.925
2	0.000	0.000	.250	.650	1.000	1.000	1.000
5	.275	.420	.700	.850	1.000	1.000	1.000
4	0.000	0.000	.850	1.000	1.000	1.000	1.000
6	.825	1.000	1.000	1.000	1.000	1.000	1.000
3	0.000	0.000	.850	1.000	1.000	1.000	1.000

INDIVIDUAL SYSTEMS

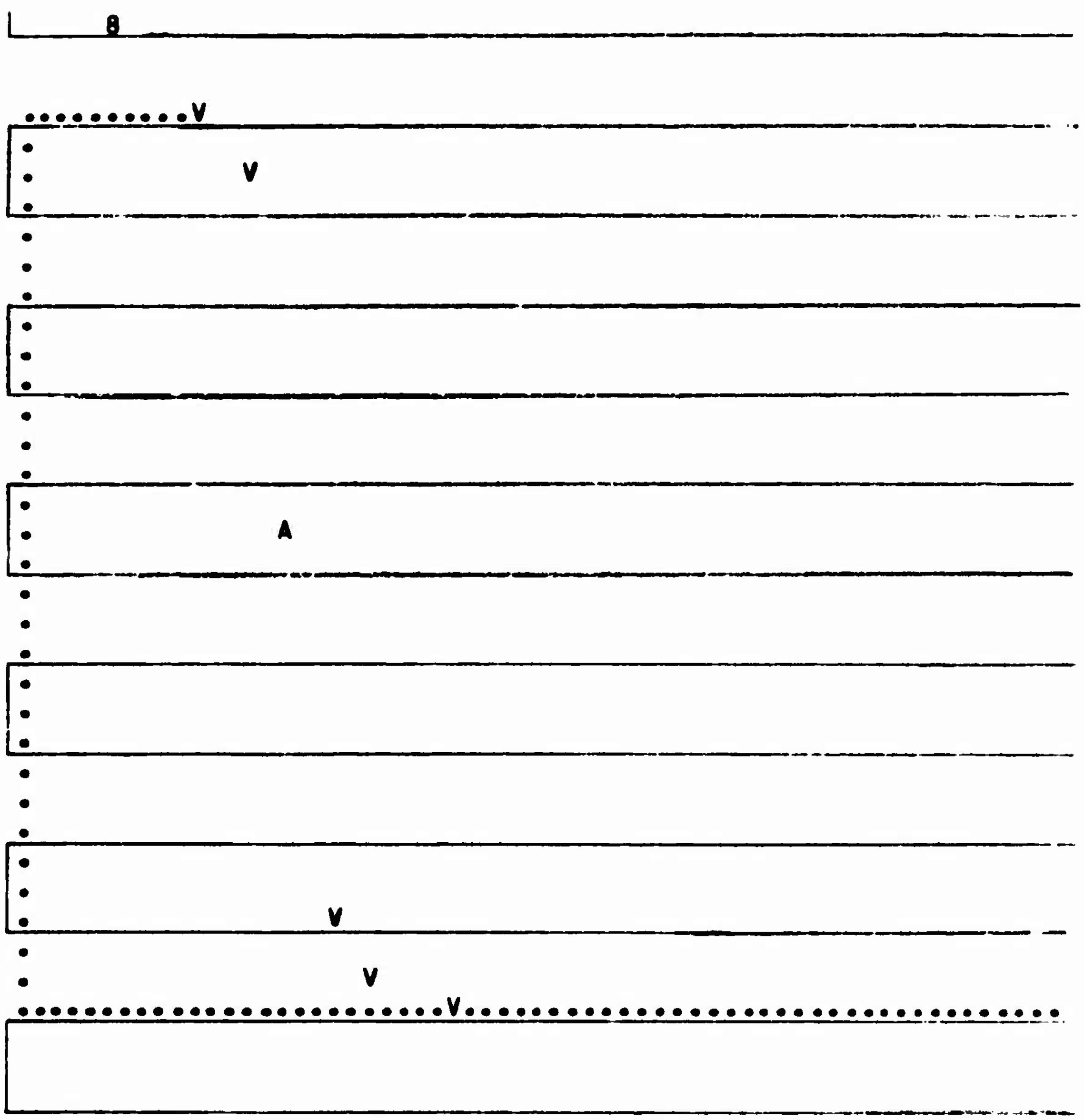
NAME	1	10	30	50	70	90	99
-100	0.000	0.000	0.000	.011	.029	.046	.375
-1	0.000	0.000	0.000	.015	.035	.117	.375

Table 4-A. Histogram Output

PS=0	(- --- HISTOGRAM (IN STEPS OF .05) FOR PS GREATER THAN 0 AND LESS THAN -----)																				PS=1	
42	0	1	1	1	0	1	0	0	1	0	1	0	0	0	0	1	0	0	0	0	1	101
24	2	0	1	2	1	1	0	1	0	1	2	1	1	0	0	2	0	1	2	3	5	102
12	0	0	0	6	0	0	0	2	0	0	0	4	0	0	0	3	0	0	0	0	23	103
35	1	0	0	2	0	0	0	0	2	0	0	0	0	0	1	0	0	0	0	0	9	601
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	49	602
5	0	0	0	5	0	0	0	1	0	0	0	4	0	0	0	6	0	0	0	0	29	603
1	0	0	0	1	0	0	0	2	0	0	0	7	0	0	0	9	0	0	0	0	30	604
13	0	1	1	0	0	2	0	0	2	1	0	0	1	2	1	4	1	0	2	1	18	705
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	48	706
1	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	46	131
5	0	0	0	6	1	0	2	5	1	1	0	3	0	0	0	5	3	1	1	0	16	132
45	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	2	133
10	2	2	1	2	0	1	2	2	0	1	1	2	2	1	1	2	1	0	1	2	14	2
4	0	0	0	3	0	0	0	1	0	0	1	1	0	1	1	2	1	0	2	2	31	3
5	1	0	1	4	0	2	1	3	3	0	1	9	2	1	1	5	0	0	0	5	6	5
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	2	3	4	5	14	20	9
12	11	1	6	4	3	5	2	1	0	1	1	1	1	0	0	0	0	1	0	0	0	10
3	8	7	3	3	3	2	2	0	0	3	3	1	1	2	1	2	1	2	1	2	0	11
2	3	1	3	3	1	1	0	3	6	1	1	1	1	4	0	5	2	6	1	5	0	12
21	7	4	1	0	1	3	1	0	2	0	0	1	0	0	3	1	0	2	1	2	0	8
37	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1
21	29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-100
21	7	4	1	0	1	3	1	0	2	0	0	1	0	0	3	1	0	2	1	2	0	8
2	35	5	3	1	1	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	2	3	4	5	14	20	6
0	2	0	0	1	1	0	0	2	3	0	2	5	2	3	0	5	1	0	1	5	17	5
13	0	2	1	0	0	2	0	0	2	1	0	0	1	2	1	4	1	0	2	1	17	4
4	0	0	0	4	0	0	1	1	0	0	0	4	0	0	1	5	1	0	0	0	29	3
5	1	1	0	5	1	0	2	4	1	1	0	3	0	0	0	5	2	2	1	0	16	2
44	1	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1

The process described on page 4-3 is repeated for each overpressure (OP) or miss distance (R) designated in input.

For each system and subsystem designated (components are not plotted), a printer plot is displayed if "PLOTON" and "PLOT$" were input. Again, the plots are identified by the "minus" number for system and non-signed number for subsystem. The abscissa is the overpressure. The ordinate is the probability of survival from 0 to 1.0. The points are designated at the vertex of the letters "A" and "V". The following example shows subsystem 8 plotted.



4.2 ERROR MESSAGES

The program does extensive input checking for errors or inconsistencies before allowing the job to go into execution. The various error messages produced along with an identification of the subroutine from which they are printed, and some reasons for their occurrence are listed below.

PROGRAM COULD NOT IDENTIFY VARIABLE NAME.

FATAL ERROR PRECEDES COLUMN XXX CARD XXX.

ERROR SCAN CONTINUES WITH EXECUTION SUPPRESSED.

Printed from: NLNAME

Explanation: Look for (1) missing commas or (2) other format errors, or (3) misspelled word (name).

FATAL ERROR - EXPECTED INTEGER NUMBER PRECEDING COLUMN XXX CARD XXX

Printed from: NLINTG

Explanation: Look for (1) missing commas or (2) non-integer number.

COULD NOT RECOGNIZE LOCAL ENV. RELATION.

FATAL ERROR PRECEDES COLUMN XXX CARD XXX.

ERROR SCAN CONTINUES WITH EXECUTION SUPPRESSED.

Printed from: COMPNT

Explanation: The instructions for finding component response from environment came across a format error - usually, a mispunched "Ri", "CAN" or "USER".

TOO MANY ENV. RELATIONSHIPS FOR INSC TABLE.

FATAL ERROR PRECEDES COLUMN XXX CARD XXX.

ERROR SCAN CONTINUES WITH EXECUTION SUPPRESSED.

Printed from: COMPNT

Explanation: Too many calls for component response operations - redimension INSC.

EQUATION TOO LONG FOR KEY - DIVIDE EQUATION.

FATAL ERROR PRECEDES COLUMN XXX CARD XXX.

ERROR SCAN CONTINUES WITH EXECUTION SUPPRESSED.

Printed from: BOOLE

Explanation: Network expression too long - watch for extraneous parentheses. Note: *operation has precedence over + operation.

NAME NOT FOUND OR SYMBOL ERROR.

FATAL ERROR PRECEDES COLUMN XXX CARD XXX.

ERROR SCAN CONTINUES WITH EXECUTION SUPPRESSED.

Printed from: BOOLE

Explanation: Watch for (1) mispunch or (2) missing component card, subsystem, or system network.

DIMENSION PS ARRAY LARGER FOR BOOLEAN SOLUTION.

FATAL ERROR PRECEDES COLUMN XXX CARD XXX.

ERROR SCAN CONTINUES WITH EXECUTION SUPPRESSED.

Printed from: LOGIC

Explanation: Network has too many parentheses - redimension PS array. Note: Do not place parentheses around single term.

TOO MANY INSTRUCTIONS FOR INSNET.

FATAL ERROR PRECEDES COLUMN XXX CARD XXX.

ERROR SCAN CONTINUES WITH EXECUTION SUPPRESSED.

Printed from: LOGIC

Explanation: Too many BOOLEAN OPERATIONS (*,+)

ENVIRONMENT XXX IS MISSING. EXECUTION SUPPRESSED.

Printed from: SELECT

Explanation: At least the "MODE" for the environment was not given - the user should check all environment parameters.

FATAL ERROR - $\left\{\begin{matrix} \text{C} \\ \text{SS} \\ \text{SYS} \end{matrix}\right\}$ XXX DELETED. CHECK NETWORK

$\left\{\begin{matrix} \text{SS} \\ \text{SYS} \end{matrix}\right\}$ XXX

Printed from: NETCK

Explanation: Component, subsystem or system missing but referenced in subsystem or system.

WARNING - COMPONENT XXX IS NOT REFERENCED.

Printed from: NETCK

Explanation: The component is defined by a component card but none of the network equations referenced the component.
Note: Non-fatal error, execution proceeds.

SYS XXX USES $\left\{\begin{matrix} \text{C} \\ \text{SS} \\ \text{SYS} \end{matrix}\right\}$ XX XXX TIMES . NETWORK

PROBABILITY IS INCORRECT.

Printed from: NETCK

Explanation: Component, subsystem, or system is used more than once in basic system expression.

FATAL ERROR - HOB TOO LARGE.

Printed from: PTOR

Explanation: Height of burst is too large for the range and/or yield of construction.

PARENTHESES DO NOT BALANCE

Printed from: BOOLE

Explanation: Self-explanatory

5.0 COMPUTER PROGRAM

This section and Appendix B describe the FAST computer program. Appendix B contains a listing of the code which includes comment cards. This section covers aspects relating to the logic, structure, and operating requirements of the entire program.

5.1 PROGRAM LOGIC

The program logic is described by its natural division into initialization and reinitialization, data input, preprocess and error checking, computation, and output. Table 5-A shows how the FAST routines divide into the various functions. In addition, Figure 5-1 outlines the logic flow or calling sequences and identifies which subsection (of 5.1) describes each calling sequence.

As Figure 5-1 shows, the program begins with the routine called MAIN, which calls subroutine NLREAD. During execution, NLREAD drives the program logic by initiating calls to various routines or sequences of routines. When a STOP$ card is encountered at the end of the data, NLREAD returns to MAIN where the execution is terminated.

The logic in NLREAD is primarily controlled by the names, like STOP$, that are punched on the data cards. These names as well as data names are described in Section 3.0 and can be seen in the listing of MAIN. MAIN stores the names in common block/NLIST/ which is passed to subroutine NLREAD.

The following description of the program logic assumes the reader will refer to or be familiar with the input (particularly the input names), the listing of NLREAD, Table 5-A and Figure 5-1.

5.1.1 Initialization and Reinitialization

The program execution begins in MAIN, which initializes tables and variables that need to be set only once for a computer run (MAIN serves in place of a block data routine since this feature is not available on all computers). The rest of the variables are initialized by subroutine RESET, which is automatically called, by NLREAD, before the program reads the first job.

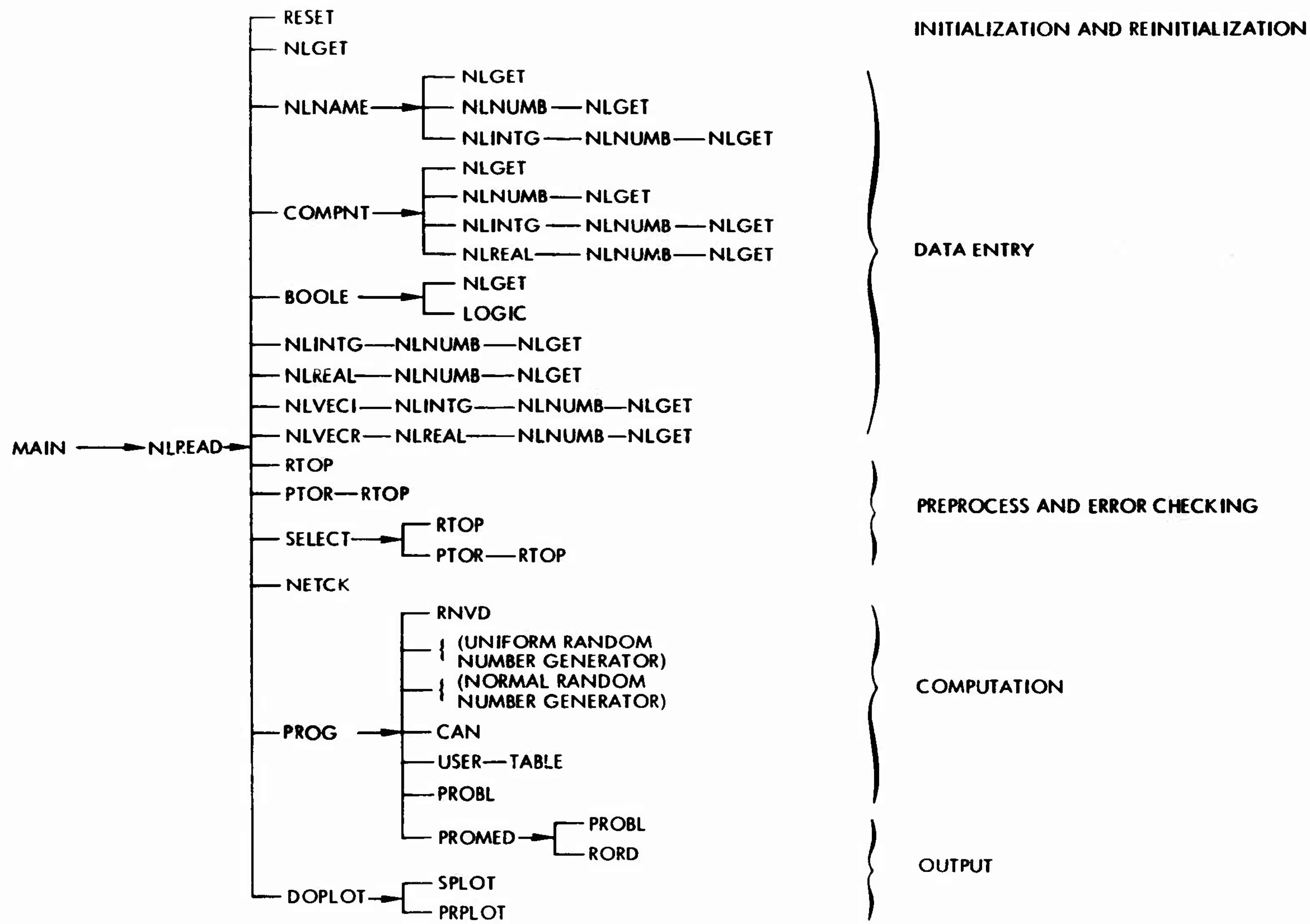


Figure 5-1. Hierarchy of Subroutine Calls

Table 5-A. Abstracts of FAST Routines

Initialization and Reinitialization	
MAIN	Interfaces program with different computer systems, dimensions program vectors and matrices, and serves as a block data routine to initialize tables and variables that are set only once for a computer run.
RESET	Initializes those variables that must be set before each job in a computer run.
Data Entry	
NLREAD	Controls the program flow by calling the appropriate routines for each input name read by NLNAME.
NLGET	Reads and prints 80 column data cards. NLGET also provides the card character recode option.
NLNAME	Identifies input names (read from data cards) and converts subscripts on the cards to integer numbers.
NLNUMB	Converts card numbers to signed integer, fraction and exponent. NLNUMB also controls how many times the number is to be repeated, identifies when NLREAL should take log base 10 of the number, and whether the next card field contains a number or input name.
NLREAL	Forms and stores real (floating point) numbers read by NLNUMB.
NLINTG	Forms and stores integer numbers read by NLNUMB.
NLVECR	Repeatedly calls NLREAD to enter a vector of real numbers. NLVECR indexes and counts the vector elements.
NLVECI	Repeatedly calls NLINTG to enter a vector of integer numbers. NLVECI indexes and counts the vector elements.
COMPNT	Interprets and stores component card information. COMPNT is also used to delete a component for case stacking.

Table 5-A. Abstracts of FAST Routines (Continued)

Data Entry (Continued)	
BOOLE	Stores a complete network equation from the equal (=) symbol to the dollar ($) symbol; translates the component, subsystem, and system names to addresses; and calls subroutine LOGIC. BOOLE is also used to delete equations for case stacking.
LOGIC	Translates the network equation into a symbolic table of instructions for finding the probability of failure.
Preprocess and Error Checking	
RTOP	Converts ranges to equivalent pressures.
PTOR	Converts pressures to equivalent ranges.
SELECT	Selects from the data base those environments required for running a case and scales the ULN to yield and height of burst.
NETCK	Checks the network table of instructions for possible errors.
Computation	
PROG	Performs the Monte Carlo assessment of component, subsystem, and system probability of failure.
RNVD	Decomposes the covariance matrices.
CAN	Computes the sum of environments as part of the transfer function. This routine will sum log base 10 environments by taking antilogs (this routine cannot combine log and non-log environments).
USER	As part of the input, the user can provide this routine for azimuth sensitivity.
TABLE	If the azimuth data is in tabular form USER can call TABLE - no other FAST routines call TABLE. TABLE performs a table look-up with linear interpolation.

[Included in the computation section are two random number generators - one for uniformly distributed numbers and one for normally distributed numbers. These routines are not identified by name because the names will be different on other computer systems. See comment cards in subroutine PROG.]

Table 5-A. Abstracts of FAST Routines (Continued)

<u>Output</u>	
PROMED	Prints the histogram and processes the histogram to find the median and other confidence levels for printing. The medians are optionally saved for DOPLOT.
PROBL	Used by both PROG and PROMED to integrate histogram data to find median and confidence levels for convergence tests and printing.
RORD	Reorders the components, subsystems, and systems by hardness of the median values.
DOPLOT	Forms plot borders and controls plotting operation.
SPLOT	Determines plot symbol and position for each point on plot page.
PRPLOT	Prints lines with two plots per page.

A run may contain more than one job where each job is like a separate computer run. A RESET$ card identifies the beginning of a new job and causes NLREAD to again call subroutine RESET.

There is one important difference between separate computer runs and stacked jobs - the starting values (NUMUR and NUMNR) for the random number generators are not reset for each job. This means that successive jobs in a run use different sequences of random numbers, whereas separate runs start with the same sequence each time. However, the RANUN data card may (on some computer systems) be used to set or reset NUMUR and NUMNR.

There are some other variables that are not reset in job stacking. These variables, which are affected by CONFI, PLOTON and RECODE data cards, can be found in the listing for MAIN under the comments "SET DEFAULT VALUES" and "INITIALIZE READING OF CARDS BY NLGET". Generally, all jobs in a run will use the same conditions so not resetting these variables is not of concern.

Job stacking should not be confused with case stacking. The RESET$ card purges all previous information, but case stacking only replaces or removes selected information, which is done in the data entry routines.

5.1.2 Data Entry

Once the program has been initialized by MAIN and RESET, NLREAD calls NLGET to read the first data card. NLGET is the only routine that reads cards, and is the only routine that recognizes and processes the RECODE card (the RECODE card is not returned as a data card). Before an 80-character data card is printed and returned in the 80-word vector called ICARD, the characters are recoded if the recode option is in effect.

The call to NLGET is within the first of two major loops in NLREAD and is executed after a RUN$ or PLOT$ card, where it is necessary to print other information before the next data card is read and printed. If in the future it is necessary to read fixed format data, a new name$ could be introduced; then following the name$ would be the fixed format cards which would be processed before a new call to NLGET.

The second major loop in NLREAD begins with a call to NLNAME. Since the cards are punched in free-field format, the data are treated as a sequence of logical records where each record begins with an identifying input name. NLNAME is used to translate each name into an integer code NAMEIX, which is used by NLREAD to branch to a section of code to process the logical record.

The data entry routines process logical records that contain a list of information to be read and stored. The forma for two of the logical records is sufficiently complex to require special routines - NLREAD calls COMPNT to process the component cards and BOOLE to process the network cards. The format for the rest of the data is embedded in NLREAD which makes use of the generalized routines NLVECI, NLVECR, NLINTG and NLREAL (see Table 5-A).

Subroutine NLNUMB is used to read numeric fields on the cards. Every field is assumed to have the following format:

$$r * \pm i \,.\, f \,{}^{E}_{L} \pm eT$$

NLNUMB stores the various parts of this format in common block/CARDAT/where NLINTG and NLREAL form the parts into a number.

r* The * identifies and separates r from the rest of the field. The r is stored as IREP and is the number (integer) of times NLNUMB will be called before NLNUMB formats the next field (3 * 4$ is the same as 4, 4, 4 $).

±i . f The positive (the + is ignored and need not be punched) or negative number is stored in two parts in INTPRT = (± i) and FRCPRT = (± · f). The number of digits in i and f are added together and stored in NULCNT. If NULCNT is non-zero, then NLINTG or NLREAD will form and store the new number (in place of a number set by MAIN or RESET or entered on a previous card). If NULCNT = 0 then the number which was read in a previous case or on some other cards is not replaced.

${}^{E}_{L}$ ± e The E and L separates the exponent from the number and the positive or negative integer value of e is stored in IEXPRT. Again, the + does not need to be punched. If X represents the number, then E means $X*10^{\pm e}$ where ILOG will be set to zero. The L means to take the log base 10 of $X*10^{\pm e}$ or ±e + log 10 (X), where ILOG = 1.

T T terminates or separates one field from another. T is interpreted by NLNUMB and represented by the integer code NEXT. Since NEXT is a complicated parameter to define, the conditions are illustrated in Table 5-B.

Table 5-B. Illustration of Variable NEXT

RULES

NEXT=1 two numbers separated by a comma

NEXT=2 two numbers separated by slash - used to terminate a vector

NEXT=3 number followed by dollar symbol

NEXT=4 number followed by namelist name or some other special symbol

CARDS

ALPHA=.4,.6,BETA=.3

.2MODE=2*1

C101=/0,10/1,20/R1=3$

DESCRIPTION

ALPHA=.4 (NEXT=1 - two numbers separated by comma)

.6, (NEXT=4 - number followed by name where comma is ignored)

BETA=.3 (NEXT=1 - comma assumed because next card starts with a number)

.2 (NEXT=4 - number followed by a name)

MODE=2*1 (same as 1,1 so NEXT=1 and then NEXT=4 after a second value because it is followed by a name)

C101=/0,(NEXT=1) 10/ (NEXT=2 - numbers separated by slash)

1, (NEXT=1) 20/ (NEXT=4 - slash ignored because non-numeric R found. The R is read by subroutine COMPNT.)

R1=(NEXT=4 - non-numeric equal read by COMPNT)

3$ (NEXT=3 - number followed by dollar symbol)

There is an execution inhibit flag (INHIBT) in common block /CARDAT/ that is incremented (for each error) if the program finds an error in the data. This flag prevents the program from running the job and machine time is not wasted by trying to perform the Monte Carlo assessment. If a RESET$ card is found, subroutine RESET turns the flag OFF (INHIBT = 0), because the next job will have a completely new set of information. In order to continue scanning the data NLNUMB is used to skip numeric fields until the next name is found, which starts the next logical record.

5.1.3 Preprocess and Error Checking

The RUN$ card indicates that all of the data has been read for a case. At this point NLREAD checks the inhibit flag. If it is non-zero (which means there is an error) NLREAD transfers to the major loop (in NLREAD) that reads the next data card and the program remains in the data entry procedure. If the inhibit flag is zero then NLREAD calls the preprocess and error checking routines SELECT and NETCK before calling PROG to perform the Monte Carlo assessment. Since SELECT and NETCK can also set the inhibit flag, the flag is again checked before calling PROG and the same transfer takes place as was just described.

FAST has an environment "data bank" capability. This means that the user may enter more environments than are required for the case. Since the component cards identify which environments are required, subroutine COMPNT maintains a list of the environments in the partitioned vector called MODE by adding a one to an element of MODE each time the environment is requested. If a component is deleted or replaced, COMPNT subtracts one from the appropriate elements of MODE.

Subroutine SELECT scans the MODE vector to determine which environments are required and then condenses the environment information (particularly the covariance matrix and correlation matrix). SELECT also scales the input ULN from the nominal yield and height of burst to the yield and height of burst of interest.

Both SELECT and NLREAD call the range to pressure (RTOP) and pressure to range (PTOR) routines. This allows the user to enter either ranges or pressures and the program will compute the equivalent values, since both are required by the program.

NETCK scans the Boolean networks for errors. Several error messages can be printed which are described in Section 4.1 and can be found in the NETCK listing.

5.1.4 Computation

If no errors were found in the data, a RUN$ card causes NLREAD to call subroutine PROG to perform the Monte Carlo assessment. Although PROG is called, the program returns to NLREAD if RNVD finds that a covariance matrix is not positive semidefinite (see Section 2.2.5.2).

The reason RNVD is called from PROG rather than SELECT, is that the K-factor uncertainty for log base 10 environments is dependent on the scaled mean. Thus, the uncertainty covariance must be constructed from a new K-factor each time the ULN is called. Log Base 10 environments are not commonly used, so this process is not discussed in the text although a programmer could determine what is done from the PROG listing.

PROG performs the Monte Carlo assessment of component, subsystem and system survivability. This process is described by comment cards in the PROG listing showing the major loops and the purpose of each section of code. Because the program is designed to simplify redimensioning variables, which is discussed in Section 5.2.1, the way information is stored and indexed is somewhat complicated.

It is beyond the scope of this manual to trace each variable through the program and show how it is used, but the following information may be useful. Section 5.2.2 describes the common blocks as to how most of the data is stored and what information the blocks contain. The environment data is read by NLREAD, rearranged in SELECT and used by PROG. The component cards, including fragility curves and transfer functions, are read by COMPNT and used by PROG. Since component cards may be deleted or replaced, they are not necessarily stored in

the same order as they are read, but can be located by indirect indexing using common block /CC01/. The same is true of the network data, which can be found using common block /CNC2/. The network is read by BOOLE and the translated instructions are stored by LOGIC in common block /CN01/ which is used by PROG.

5.1.5 Output

Subroutine PROMED, which is called by PROG, prints the histogram and confidence matrix. These results are printed for each range/pressure that is read from either the R or OP data cards. If the plot option (PLOTON) has been requested, PROMED stores the 50% values in the matrix PROB of common block /CT07/, which is used by the printer plot routines.

NLREAD calls DOPLOT when a PLOT$ card is encountered in the data. Since cases may have to be stacked to change the scaling parameters as the value of pressure increases, the PLOT$ card may appear after the last case and causes all of the results to be plotted as a single plot. This is accomplished since common block /CT07/ can contain results from several RUN$ cards.

In order to have a machine independent plot capability, the printer is used to plot the data. The plots are sufficiently accurate unless the available printer produces wavy lines. Currently only the 50% curves for subsystems and systems are sorted by PROMED and plotted by DOPLOT, SPLOT and PRPLOT.

5.2 PROGRAM STRUCTURE

The program is structured to provide maximum problem solving flexibility to the user while maintaining ease of usage. On many computers it is cost effective, with a long running Monte Carlo program, to use a minimum amount of core memory. For this reason, care has been taken to simplify the redimensioning of variables when unusually large or complex problems must be solved. FAST is used on various computers so a special emphasis has been placed on making the program as machine independent as possible. These features, along with common block organization, are discussed in the following sections.

5.2.1 Variable Redimensioning

Most variables are completely dimensioned only in the main driver, and are dimensioned as dummy, one-word vectors in all other routines. Dimensions depend on the size of the problem to be solved, such as the number of components, the number and complexity of network equations, or the number of environments. In redimensioning the program, it is necessary to make sure that the maximum problem size is properly set by MAIN and that interrelated as well as directly related dimensions are changed. Table 5-C and the listing for MAIN better illustrate what needs to be done, and if a mistake is made there are some error messages that may be helpful (see Section 4.2).

The technique used to simplify redimensioning makes use of properties of common blocks. As the program is loaded, the loader allocates core by the maximum dimension of the common block in MAIN. In the subroutines the blocks appear shorter but the starting address and amount of available core are the same. The dimensioned variable is always the last parameter of the block. It should be noted that the variables IPAD1, IPAD2 and IPAD3 are not used by the program, but are used to force real variables to even word boundaries for the IBM 360.

5.2.2 Common Blocks

The letters in the common block names help to identify what kind of information is stored in the block (see Table 5-C). Table 5-D cross references common blocks to subroutines. As mentioned in Section 5.2.1, Table 5-C indicates how the blocks are dimensioned.

The environment common blocks are more complex than the others, because the program has an environment data bank capability. The read routines store all the possible environments in an alternate region. Then, depending on which environments are called for by the component cards, subroutine SELECT copies those environments to the primary region prior to the Monte Carlo iterations. The data bank capability reduces the chance that an environment will be missing, and at the same time, it keeps the program from performing unnecessary computations. The MODE vector, which is partitioned into four parts, contains the selection information and is used by subroutine SELECT.

Table 5-C. Common Blocks - Contents and Dimensions

/Block/ and Variable	Dimension	Comments
/WORKEY/		Primarily used by BOOLE and LOGIC in processing network equations.
IKEY		Initial (or indexed) word of network equation in WORK (IKEY).
LKEY		Last word of information.
MKEY		Maximum length of WORK. MKEY $\geq$ MNAME.
IPAD1		Used only to pad real variable to even word boundary for IBM360.
WORK	MKEY	Used by BOOLE and LOGIC and as a temporary scratch by various routines.
/IOUNIT/		Identifies FORTRAN logical input/output units.
IREAD		Card input.
IPRINT		Printer output.
IPUNCH		Not currently used.
ITAPE		
/CARDAT/		Information used by data entry routines.
NCARD		Contains number of cards read by NLGET - printed with error messages.
IECHO		Logical output file where NLGET copies input cards - generally IECHO = IPRINT.
INHIBT		= 0 No input errors; $\neq$ 0 inhibit Monte Carlo assessment.
NCODE		= 0 No recode: $\neq$ 0 recode card characters (used only by NLGET).
NULCNT		= number of integer + fraction digits - store new number; = 0 use previous value.
NEXT		Indicates whether the next field is a number or an input name.
IREP		Number of times to store a value.
INTPRT		Integer part of a number.
FRCPRT		Fractional part of the number.

Table 5-C. Common Blocks - Contents and Dimensions (Continued)

/Block/ and Variable	Dimension	Comments
ILOG		= 0 Do not take log; = 1 take log base 10 of input number.
IEXPRT		Exponent of number.
ICOL		Card column index.
LCOL		Last card column with information.
MCOL		For current program MCOL = ICOL.
ICARD	80	Contains 80-column input card.
/CHRLST/		
KSET	47	A through Z, numbers and special FORTRAN BCD character set.
/COMMON/		
NLIST	219	Name index, number of characters and characters of input names used by NLNAME (NLIST ends with two zeros)
/CE01/		
NBYN		Actual number of environments selected for a case j = 1, ..., NBYN
MBYM		Maximum number of environments the program can accept j = 1, ..., MBYM
MODE	MBYM, 4	MODE is partitioned into 4 parts: MODE(j, 1) selected values for case MODE(j, 2) input from MODE or ENV cards - should be zero if environment not entered MODE(j, 3) = 0 if environment not required, otherwise contains the number of times the environment was requested MODE(j, 4) contains the name of only the requested environments
/CE02/		
ULN	MBYM, 2	ULN(j, 1) Selected values for case ULN(j, 2) Values from ULN or ENV cards

Table 5-C. Common Blocks - Contents and Dimensions (Continued)

/Block/ and Variable	Dimension	Comments
/CE03/ FACTK	MBYM, 2	FACTK(j, 1) Selected values for case FACTK(j, 2) Input from K-FAC or ENV cards
/CE04/ ALPHA	MBYM	Input from ALPHA or ENV cards (since yield sealing is done in SELECT there is no alternate region)
/CE05/ BETA	MBYM, 2	BETA(j, 1) Selected values for case BETA(j, 2) Input from BETA or ENV cards
/CE09/ GAMMA	MBYM	Input from GAMMA or ENV cards (no alternate region - scaling in SELECT)
/CE09A/ HB	MBYM	Input from HB-GAMMAB or ENV cards (no alternate region - scaling in SELECT)
/CE09B/ GAMMAB	MBYM	Input from HB-GAMMAB or ENV cards (no alternate region - scaling in SELECT)
/CE06A) SIGSS	$\frac{MBYM*(MBYM-1)}{2}$	Selected values form SIGSSS
/CE06B) SIGSSS	$\frac{MBYM*(MBYM+1)}{2}$	Input from S cards
/CE06C) CSS	$\frac{MBYM*(MBYM+1)}{2}$	Decomposed result of SIGSS (see RNVD)
/CE06D/ IXSS	MBYM	Indexes environments in decomposed CSS

Table 5-C. Common Blocks - Contents and Dimensions (Continued)

/Block/ and Variables	Dimension	Comments
/CE07A/		
SIGAU	$\frac{MBYM*(MBYM+1)}{2}$	Constructed (in PROG) from FACTK(j, 1) and RHOCOS correlation matrix
/CE07B/		
RHOCOL	$\frac{MBYM*(MBYM+1)}{2}$	Selected values from RHOCOS
/CE07C/		
RHOCOS	$\frac{MBYM*(MBYM+1)}{2}$	Input from A cards
/CE07D/		
CAU	$\frac{MBYM*(MBYM+1)}{2}$	Decomposed from SIGAU (see RNVD)
/CE07E/		
IXAU	MBYM	Indexes environments in decomposed CAU
/CE08/		
RNVDX	MBYM, MBYM	Used by RNVD to decompose a matrix
/CR01/		
IDOSS		= 0 do not test convergence of subsystems = 1 test
NIMIN		Do NIMIN inner loops before testing inner loop convergence
NIMAX		Stop testing inner loop after NIMAX outer loops
NOMIN		Do NOMIN outer loops before testing outer loop convergence
NCONFI		Number of elements in CONFI vector (0 < NCONFI < 8)
NUMUR		Starting value for uniform random number generator
NUMNR		Starting value for normal random number generator
EPSI		Epsilon for inner loop convergence
EPSO		Epsilon for outer loop mean convergence
EPSOC		Epsilon for outer loop CONFI convergence
EPSMAT		Epsilon used in RNVD to test for zero
CONFI	7	Generally only 50% (.5) confidence is tested

Table 5-C. Common Blocks - Contents and Dimensions (Continued)

/Block/ and Variables	Dimension	Comments
/CR02/		
WO		Nominal yield input on NOMRWH or NOMPWH card
W		Yield of interest input on W card
HO		Nominal HOB input on NOMRWH or NOMPWH card
H		HOB of interest input on H card
OPO		Nominal pressure input on NOMPWH card or computed by RTOP
RO		Nominal range input on NOMRWH card or computed by PTOR
R	20	Ranges of interest entered on R card or computed by PTOR
/CR03/		
NR		Number of ranges or pressures entered
NOP		= 0 indicates ranges entered (is later set to NR); = NR indicates pressures entered
OP	20	Pressures of interest entered on OP card or computed by RTOP
/CNC1/		
NCR		Actual number of components read (indexed 1 to NCR)
NSYS		Actual number of subsystems and systems read (indexed MNAME-NSYS+1 to MNAME)
MNAME		Maximum number of components + systems + subsystems
NAME	MNAME	Names of components, subsystems, and systems
/CNC2/		
INSLOC	2, MNAME	Used to locate component, subsystem, and system information.
/CNC3/		
MPS		Maximum length of P vector where MPS > MNAME
IPADZ		Used only to pad P to even word boundary for IBM 360
P	MPS	i = 1, MNAME contain probabilities for compoments, subsystems, and systems. Excess cells used to resolve network probability.

Table 5-C. Common Blocks - Contents and Dimensions (Continued)

/Block/ and Variables	Dimension	Comments
/CC01/		
MCINS		Actual number of instruction
MCINS		Maximum number of instructions
INSC	4, MCINS	Instructions for forming local environment from external environment
/CC02/		
COEF	3, MCINS	Transfer function coefficients
/CC03/		
FCY	6, 100	Fragility ordinate can handle up to 100 components (6 point curve)
/CC04/		
FCX	6, 100	Fragility abscissa - mode (best estimate curve)
/CC05/		
FC	6, 100	Fragility abscissa with uncertainty for inner loop.
/CC06/		
FU	6, 100	Width of uncertainty
/CC07/		
IBD	8, 100	The first 6 points define the mode or beta distribution to use IBD can be zero if a particular point does not have uncertainty. $IBD(7,\ell)$ is the number (2 through 6) of fragility break points. $IBD(8,\ell)$ will be zero if none of the points have uncertainty (best estimate curve only)
/CN01/		
NNINS		Actual number of instructions
MNINS		Maximum number of instructions
INSNET	4, MNINS	Instructions from network equations
/CT01/		
X	MBYM	XBAR with uncertainty
/CT02/		
XBAR	MBYM	Scaled ULN

Table 5-C. Common Blocks - Contents and Dimensions (Continued)

/Block/ and Variables	Dimension	Comments
/CT03/		
XDEL	MBYM	X with random variation
/CT04/		
REG	MBYM	Registers used in finding local environment
/CT05/		
PZ	MNAME	ΣP_i from inner loop to find expected P
/CT06/		
PSAVE	22, MNAME	Histogram
/CT07/		
OPINCH		Scales plot points value read from PLOTON card
NPROB		Actual number of points/plot
MPLOT		Maximum number of points/plot
PROB	MPLOT, 35	Contains points for up to 35 subsystems and systems to plot
/CBETA/		
IPAD3		Used only to pad real variables to even word boundary for IBM 360
MAXBD		Number of points in each of the 9 Beta distributions
BDMAX		Used in interpolation BDMAX is a real value = MAXBD - 1
DM	9	The mode for each distribution
BD	MAXBD, 9	Tables of beta distributions

Table 5-D. Common Block/Subroutine Cross Reference

Common Block	Subroutine																											
	MAIN	NLGET	NLNAME	NLVECI	NLVECR	NLINTG	NLREAL	NLNUMB	RESET	NLREAD	COMPNT	BOOLE	LOGIC	NETCK	SELECT	RTOP	PTOR	PROG	RNVD	PROMED	PROBL	DOPLOT	SPLOT	PRPLOT	RORD	CAN	TABLE	USER
WORKEY	X											X	X		X			X	X	X								
IOUNIT	X	X	X			X				X	X	X	X	X	X		X	X	X	X		X		X				
CARDAT	X	X	X	X	X	X	X	X	X	X	X	X	X		X			X										
CHRLST	X	X	X					X			X	X	X	X								X	X					
NLIST	X		X																									
CE01	X								X	X	X				X			X										
CE02	X									X					X			X										
CE03	X									X					X			X										
CE04	X									X					X													
CE05	X									X					X			X										
CE06A	X														X			X										
CE06B	X								X	X					X													
CE06C	X																	X										
CE06D	X																	X										
CE07A	X																	X										
CE07B	X														X			X										
CE07C	X								X	X					X													
CE07D	X																	X										
CE07E	X																	X										
CE08	X																	X										
CE09	X								X	X					X													
CE09A	X								X	X					X													
CE09B	X									X					X													
CR01	X									X								X	X									
CR02	X								X	X					X			X										
CR03	X								X	X					X			X				X						
CNC1	X								X		X	X	X	X				X		X		X						
CNC2	X										X	X		X				X										
CNC3	X												X	X				X		X		X						
CC01	X								X		X							X										
CC02	X										X							X										
CC03	X										X							X										
CC04	X										X							X										
CC05	X										X							X										
CC06	X										X							X										
CC0[illegible]	X								X		X							X										
CN01	X								X		X	X	X	X				X										
CT01	X																	X										
CT02	X																	X										
CT03	X																	X										
CT04	X																	X								X		
CT05	X																	X										
CT06	X																	X		X								
CT07	X								X	X										X		X						
CT07B																						X		X				
CBETA	X																	X										
AZMUTH																		X										X

5.2.3 Computer Conversion

Program conversion to other computers is facilitated by requiring changes only to MAIN and subroutine PROG. All other routines are believed to be machine independent.

MAIN defines the FORTRAN logical input/output units, which are easy to change. MAIN also contains all data statements because the format may be different on some computers (e.g., the CDC 3800).

Since routines to generate random numbers (see Section 5.3) are machine dependent, the calls to the routines in subroutine PROG need to be changed. If the generators require starting values, the variables NUMUR and NUMNR can be set in the main driver or by placing a RANUN card in the data.

The program is structured to run on an IBM 360 or similar IBM computer, and even provides for the possibility that a data deck may contain cards punched on both an 026 and 029 keypunch (see the descriptions for the RECODE data card).

5.3 OPERATING REQUIREMENTS

The FAST program requires a card input unit and a printer output unit. No other input/output devices are currently required.

On the CDC computers, it is necessary to load MAIN first since this is the only routine that dimensions common block variables - this avoids truncated common blocks.

The program requires two random number generators, one that produces numbers uniformly distributed between 0 and 1, and one that produces numbers from a normal distribution with mean of zero and variance of unity.

The core storage requirements depend upon the efficiency of the loader and compiler (the TRW FUN optimizing compiler has been used

successfully). Although FAST is used on various computers, the only current core storage information is for TRW and Lawrence Berkeley Laboratory.

	CDC 6600 TRW	CDC 7600 LBL
LOAD	$36\ K_8$	$73\ K_8$
EXECUTE	$66\ K_8$	$63\ K_8$

6.0 SAMPLE PROBLEM

A sample problem has been included in this manual to provide a check case for comparing output, to accelerate the user's acquisition of insight to FAST techniques, to develop intuition as to what constitutes a reasonable output from FAST given a reasonable set of inputs, and to describe the development of FAST inputs for a "real" physical configuration.

To accomplish these purposes, FAST inputs for performing survivability studies for a military system subject to a nuclear threat have been developed. The sample problem considers only effects due to the airblast, ground shock, and crater debris environments, however, other nuclear weapon environments such as EMP, nuclear and thermal radiation, etc., could be considered within the FAST methodology. The sample problem was designed to exhibit important aspects of modeling the system and the response phenomenology in FAST and considers a hypothetical system and generic site conditions. The different aspects of developing the FAST inputs are described in varying degrees of detail, depending upon their significance in demonstrating the code capability and practical modeling considerations. In particular, the development of nuclear blast and shock environment inputs has been emphasized, with highlights on the shock spectral approach for evaluating ground shock induced damage.

The hypothetical system for the sample problem consists of an ICBM control facility shown in Figure 6-1. A number of these systems, built almost identically, are to be deployed at sites where geology is soil with a water table at least 200 feet below the surface and bedrock below the water table. The design hardness is 600 psi for a five megaton (5 MT) surface burst. The system is in an early stage of development, and the survivability of a baseline system is to be evaluated. This study will consider the system survivability as a function of overpressure for yields of 0.5 and 5 MT, and both surface contact and near-surface height of burst detonations.

To perform its mission, the control facility must be able to receive and transmit messages by using either the communications antenna or the hardened cable link (HCL). The command and control consoles, which are

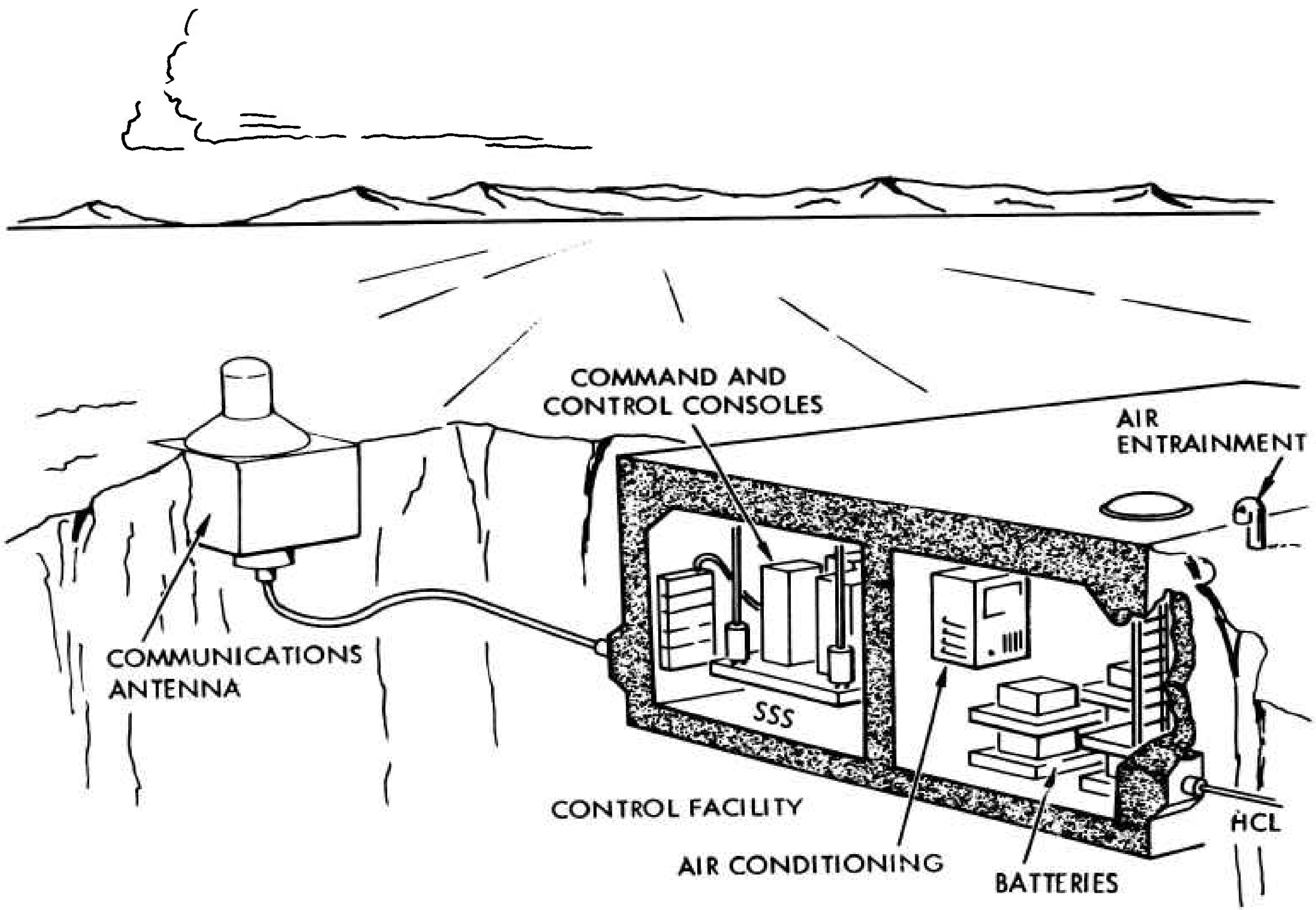


Figure 6-1. Hypothetical System

essential to mission performance, are powered by batteries and cooled by air conditioning. A sturdy rectangular structure consisting of two nearly cubical rooms protects the equipment. In normal operation, the control facility circulates air from the outside. When attacked, the blast valve closes the air entrainment system, thus protecting the control room.

Table 6-A shows a detailed hardness matrix for the system. The matrix, which relates the various functional subsystems and components with the environments to which they are most sensitive, constitutes a compact means of controlling FAST inputs. The numbers in the environment column refer to the corresponding parameters (listed in Table 6-C).

6.1 ENVIRONMENTS

This section deals with the derivation of hostile environment inputs for the FAST code sample problem. For this hypothetical system, a design threat of a surface burst of 5 MT yield at a range corresponding to 600 psi has been chosen as the baseline condition from which weapon effects predictions are to be scaled. Because of nonlinearities in scaling over large distances, the baseline threat conditions should be as near to the middle of the range of interest as possible.

Ground shock environment prediction equations, shown in Table 6-B, were developed in coordination with, and under the direction of DNA/RDA, and provide the basis for derivation of the environments. These equations consider the peak free field air induced vertical and horizontal displacements, velocities, and accelerations and the crater and related effects (CARE) displacement and velocity (where the vertical and horizontal CARE components are taken to be equal). In the equations, L is the minimum of the depths to bedrock and water table (L = 300 ft), z is the free field depth of interest, t_a is the seismic arrival time from the surface to the depth of interest, ρ is the media density, U is the airblast shock velocity, and V is the crater volume $\left(V = 50 \times 10^6 \text{ ft}^3 \left\{\frac{W}{1 MT}\right\}\right)$. The free field vertical shock spectra for the 25-foot depth and the free field horizontal shock spectra for the 15-foot depth (for the design threat) are shown in Figure 6-2. The various aspects of this spectra were designed partly to show the detail to which the spectra can be modeled and does not necessarily reflect the detail required for a given problem. The lower frequency

Table 6-A. Sample Problem Hardness Matrix

Function	Subsystem	Mission Critical Failure Mode	Response Related to Failure	Environment Parameter*	Azimuth Sensitivity
MISSION	C&C Console	Component Failure Due to H. F. Accel.	Floor H. F. SS	Vertical SS @ 2 cps (1) (5) (4)	No
				H. F. Ceiling SS (13)	No
		Base Attachment Breaks	Impact Velocity	Horizontal SS @ 0.3 cps (17) (18) (6)	Yes
	Antenna Communications	Protrusion Breaks Off	Peak Dynamic Pressure Load	Peak Dynamic Pressure (11)	Yes
	Tranceiver Element	Antenna Burial	Signal Strength	Debris Depth (19)	No
	Antenna Penetration	None (Super Hard)			
	HCL Penetration	Loss of Electrical Connectivity	Moment Loading Due to AI Vertical Shock	Stress x Relative Displacement (10)	No
SUPPORT	Battery Mounts	Mounts Break	Rel. Vert. Displ.	Vertical SS @ 10 cps (4) (3) (1)	Yes
			Rel. Horiz. Displ.	Horizontal SS @ 4 cps (6) (7) (8) (9)	No
	Battery Elements	Elements Break	Battery Acceleration	Vertical SS @ 10 cps (2) (3) (4) (8)	No
	AC Condensor	Mounting Supports Break	H. F. Structure S.S. Acceleration	Same as fragility parameter (12)	No
	Blast Valve	Delayed Closing	Closing Time	H. F. Structure S.S. Acceleration (13)	No
PROTECTIVE	Roof Structure	Structural Failure	Equivalent Static Load for Roof	Same as fragility parameter (14)	No
	Structure Walls	Structure Failure	Equivalent Static Load for Walls	Same as fragility parameter (15)	No
	Structure Footing	Structural Failure	Equivalent Static Load for Footing	Same as fragility parameter (16)	No
	SSS	Gross Failure of Floor	Rel. Vert. Displ.	Vertical SS @ 2 cps (1) (5) (4)	No

Notation:
HF - High Frequency
SS - Shock Spectra

*Shaded environments were dropped from the analysis after subsequent evaluation.
**See Table 6-C

Table 6-B. Ground Shock Environment Equations - Surface Burst

$$d_z = 2 \text{ in} \left[\frac{p}{100 \text{ psi}}\right]^{2/3} \left(\frac{W}{1MT}\right)^{1/6} \left[\frac{L}{100 \text{ ft}}\right]^{0.6} e^{-0.0015\,(z-25 \text{ ft})}$$

$$v_z = 55 \text{ ips} \left[\frac{p}{100 \text{ psi}}\right] \left(\frac{W}{1MT}\right)^{1/12} e^{-0.0015\,(z-25 \text{ ft})}$$

$$a_z = \frac{v_z}{t_a}$$

$$d_H = 1/4\ d_z$$

$$v_H = \frac{p}{\rho U} \left(\frac{W}{1MT}\right)^{1/2} \qquad 15 \text{ ft} \le Z \le 100 \text{ ft}$$

$$a_H = \frac{1}{2}\ a_z$$

$$d_{CARE} = 0.45\ \frac{V^{4/3}}{R^3}$$

$$v_{CARE} = 2.8 \text{ fps}\ \frac{(W/1MT)^{2/3}}{\left(\frac{R}{1k \text{ ft}}\right)^2}$$

response of the structure would be governed by these spectra. The median values for the baseline threat condition and the scaling laws are shown in Table 6-C. In that table, the parameters fall into four categories:

a. Free field peak air-induced ground shock displacements, velocities and accelerations

b. Derived air-induced ground shock parameters which primarily specify line segments on the shock spectra

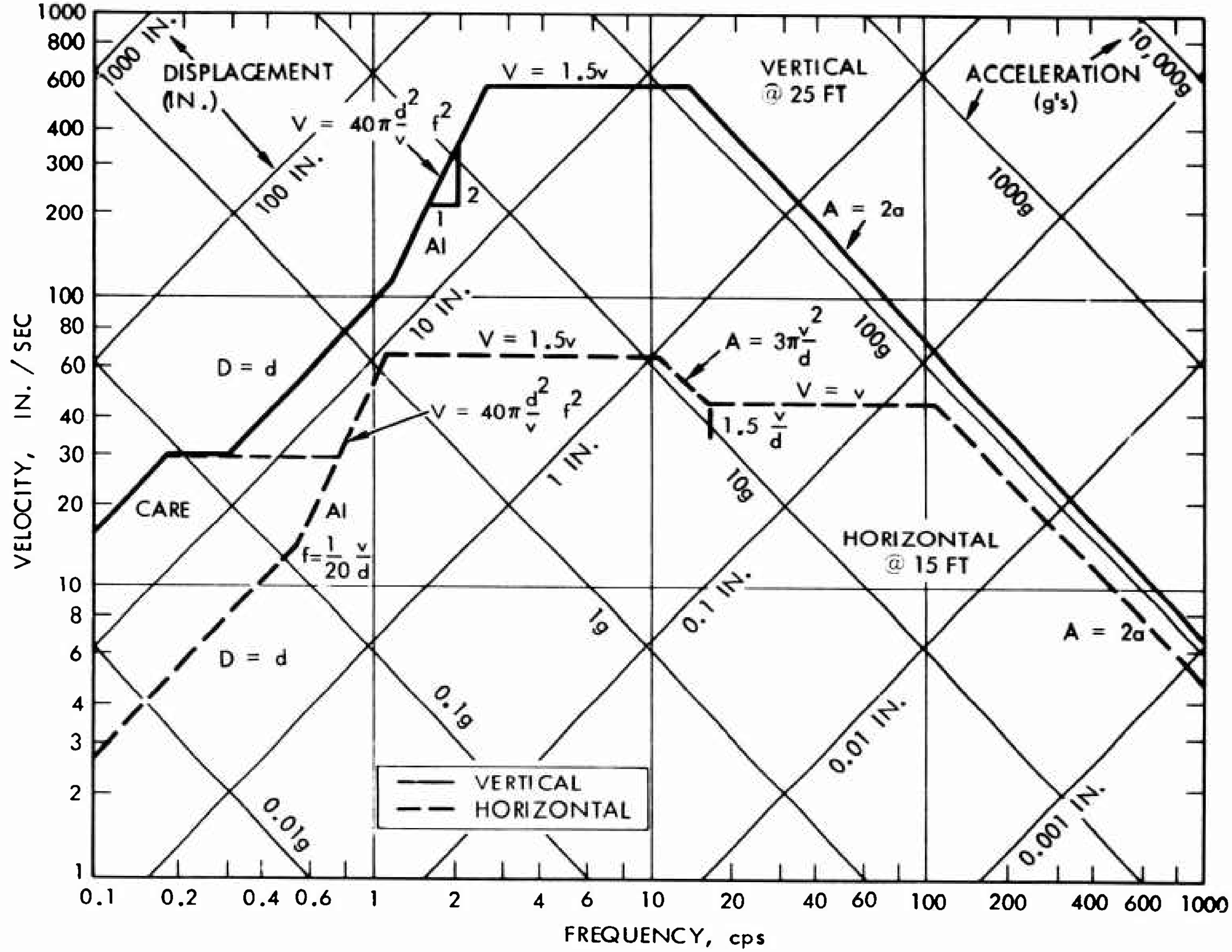


Figure 6-2. Median Shock Spectra at 5 MT, 600 psi plus Scaling Laws

Table 6-C. Median Environment at 5 MT, 600 psi plus Scaling Laws

Environment Parameter		Median Value*	Scaling			
			$E \sim R^{\beta} W^{\alpha}$		$E \sim p^{\beta} W^{\alpha}$	
			β	α	β	α
E(1)	d_z	16 in.			2/3	1/6
E(2)	v_z	380 ips			1.0	1/12
E(3)	a_z	60 g's			1.0	1/12
E(4)	$V_{z\ max}$	570 ips			1.0	1/12
E(5)	$20 d_z^2/v_z$	14 in. sec			1/3	1/4
E(6)	d_H	4 in.			2/3	1/6
E(7)	v_H	45 ips			1/2	1/12
E(8)	$V_{H\ max}$	67 ips			1/2	1/12
E(9)	$1.5\ v_H^2/d_H$	1.9 g's			1/3	0
E(10)	σd_{rel}	2400 lb/in.			1.8	1/12
E(11)	Dynamic Pressure	1400 psi			1.37	0
E(12)	a_{STR}(A/C)	100 g's			1.0	1/12
E(13)	a_{STR}(SSS)	200 g's			1.0	1/25
E(14)	Equivalent Static Roof Load	600 psi			1.0	1/25
E(15)	Equivalent Static Walls Load	600 psi			1.0	1/12
E(16)	Equivalent Static Footing Load	600 psi			1.0	1/6
E(17)	d_{CARE}	27 in.	-3.0	1.33		
E(18)	v_{CARE}	29 ips	-2.0	2/3		
E(19)	Debris Depth	12 in.	-3.86	1.6		

*5 MT, 600 psi - Surface Burst

c. Correlated air-induced effects on the control facility structure

d. Cratering and Related Effects (CARE).

Table 6-C shows that air-induced environments scale with overpressure and yield, while crater-induced effects scale with range and yield. For surface bursts on a given geology, CARE effects scale with crater volume and, thusly, with yield. For heights of burst, the crater volume reduction affects the CARE environments while changes in the overpressure/range law at different heights of burst are used to estimate changes in Pressure and Related Environments (PARE).

The list of environments in Table 6-C are designed to model the critical system response phenomenology in a statistically meaningful manner. Heading the list of weapons effects are free field vertical peak displacement, d_z, velocity, v_z, and acceleration, a_z, at the 25-foot depth below the surface. These parameters characterize the rigid-body vertical motion of the control facility. Next comes V_z, the peak vertical shock-spectral velocity (shown in Figure 6-2).

Along the displacement amplification segment of the spectrum the displacement-to-frequency ratio (D/f) is constant. The pseudo-velocity (V) must then be of the form, $V = \mu f^2$, where the proportionality factor (μ) is evaluated from the pseudo-velocity at the break point frequency (see Figure 6-2)

$$f = \frac{1}{20}\frac{v}{d} \quad \text{(in cps units)} \qquad (6\text{-}1)$$

The break point pseudo-velocity is equal to $(\pi/10)$ v and therefore the displacement amplification segment can be described by

$$V = 40\pi\frac{d^2}{v}\ f^2 \qquad (6\text{-}2)$$

The environment parameter

$$20\ d_z^2/v_z \qquad (6\text{-}3)$$

specifies the displacement amplification segment of the shock spectrum in this analysis. The horizontal peak displacement, d_H, and vertical peak velocity, v_H, in the free field at the 15-foot depth characterize the horizontal rigid-body response of the control facility. The horizontal peak shock spectral velocity is V_H. The quantity $1.5\, v_H^2/d_H$ is $1/2\pi$ times the constant acceleration line connecting the two constant pseudo-velocity lines. The quantity $\sigma\, d_{rel}$ is the critical parameter for the cable penetration response. It is the product of the backfill stress times the backfill motion, relative to the control facility. The pressure, Q, is the peak dynamic pressure associated with the passage of an overpressure shock. The structural acceleration environments are given at the ceiling attachment points of the shock isolated floor, a_{STR} (SSS), and at the location of the air conditioner, a_{STR} (A/C). Note that these environments relate more to the structural configuration than to the geological site condition. The final two classes of environments relate to structural collapse loads and to CARE environments. The CARE environments, displacement, velocity and debris, are self-explanatory. However, the structural load environments require further explanation.

The three structural subsystems (roof, walls, and footing) each have fragilities which are functions of parameters closely related to overpressure. It is important to consider correlation in the analysis of these subsystems; for example, soft foundation materials tend to cause a relatively large foundation loading and cushion the roof loading, while for the case of stiff foundation materials, the reverse is true. The key construction variables relating to the collapse of structural members include concrete strength, and workmanship in placing steel, concrete, and backfill. These construction variables also introduce correlation in the structural response. The FAST code does not permit correlation of the fragilities; however, the proper correlations between the structural subsystems, due to construction techniques and structural mechanics, can be obtained through correlation of the structural loads.

The environments giving equivalent static roof, wall and footing loads take into account the dynamic elasto-plastic responses in the failure modes of these structural components. The scaling with yield indicates

that the roof and walls which respond in shorter times than the footing are less yield-dependent than the footing, i.e., the footing load must be sustained for a longer time in order to produce collapse. Therefore, the footing response is more yield-dependent. The impulse, $I = \int_0^{t^*} P(t)dt$, up to the critical time for structural collapse, t*, is the key variable and is accounted for in the yield scaling factor.

Now that all the environment parameters have been addressed, it is worthwhile to further discuss the advantages of the shock spectrum approach in hardness analysis. The use of shock spectra greatly simplifies the prediction of the damage potential of various ground shock time histories. The shock spectrum of a shock waveform is the peak response to the waveform of a simple oscillator as a function of the oscillators' frequency. Figure 6-2 shows two shock spectra. The usefulness of shock spectra in damage prediction stems from:

a. The peak response, for many structural/mechanical systems, can be determined directly from the shock spectra.

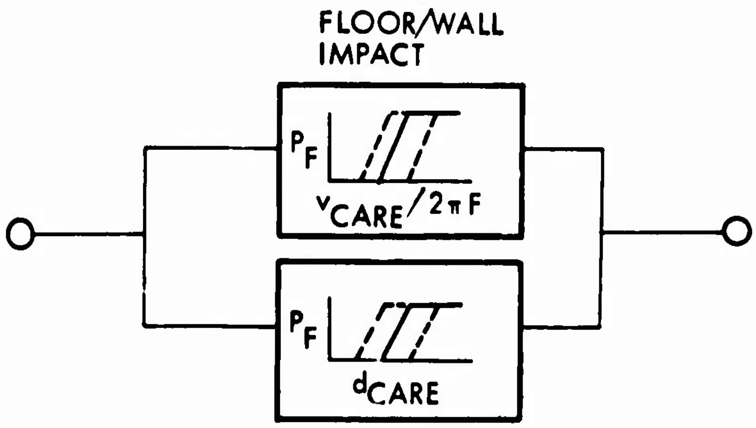
b. Shock waveforms of the same class often will have much the same shock spectra even though the actual time histories differ considerably.

Given the orientation of a line on shock spectra paper, only one parameter is needed to specify the line. For example, the solid line segments below 0.2 cps in Figure 6-2 are lines of constant maximum relative displacement, D; the lines above 100 cps are lines of constant maximum acceleration response, A; constant pseudo-velocity, V, lines are horizontal. For any point on this type of paper, D, V and A, as well as frequency, are specified. Over the years engineers have, for certain classes of pulses, used simple but accurate rules for estimating shock-spectral displacement, D = d, shock-spectral peak pseudo-velocity, V = 1.5v, and shock-spectral acceleration, A = 2a, where capital letters are spectrum parameters and small letters are free field values of peak displacement, velocity and acceleration. In general, the shock response spectrum values are amplified over peak free field values. More recently, other classes of waveforms have been investigated, implying the possible addition of the other line segments shown in Figure 6-2.

Of particular interest are the low frequency Crater and Related Effects (CARE) which, in Figure 6-2, dominate horizontal response below 1/2 cps and the ground roll frequency band in which the oscillatory nature of the groundshock waveform interacts with shock isolation systems, causing amplification of peak relative displacement. The CARE region portion of the spectrum which depends on the crater volume is equal in value for vertical and horizontal motions, and can disappear for non-cratering bursts. All these segments are related to peak free field ground shock predictions and the FAST code will select random values for these variables. The proper way to simulate all these regions of shock response is to input the entire spectrum for the FAST calculation and let the computer determine the shock spectrum segment which dominates at the various frequencies of interest.

As an example for a 600 psi surface burst, consider the shock spectrum as applied to the control facility's shock-isolated floor. The horizontal natural frequency of the floor is 0.3 cps. For 5 MT, this frequency is in the CARE pseudo-velocity region of the horizontal shock spectrum. However, for the low yields, at the same pressure, 0.3 cps would be in the CARE displacement region.

The d_{CARE} and v_{CARE} are random variables, and a bivariate sample of pairs of their values will be generated during a FAST calculation. The bivariate distribution will contain some cases where the horizontal floor response is governed by the d_{CARE} leg and other cases where the floor response is governed by the v_{CARE} leg (on a velocity leg, maximum relative displacement is inversely proportional to the frequency, i.e., $d_{rel} = V/2\pi f$). One way to select the proper value for maximum relative displacement is to create a logical subnetwork as shown below.



This network will count floor damage only if the smaller of d_{CARE} and $v_{CARE}/2\pi f_h$ is large enough to cause damage. Note that the concave downward intersection of d_{CARE} and v_{CARE} lines results in parallel logic, while a concave upward intersection between v_{CARE} and the displacement amplification line would require a series subnetwork.

For height of burst cases, the CARE environments will be small, and the shock suspension system horizontal response will be governed by the air induced displacement. The situation where the CARE and PARE environments are of the same order can also be modeled by the FAST code.

By combining shock spectra line segments in series and parallel it is possible to model an entire shock spectrum. However, for the sample problem system, soft-mounted items respond only to the lower end of the spectrum, hard-mounted items respond only at the upper end of the spectrum, and the batteries, which respond to the mid-frequencies, have been determined to depend only on shock-spectra velocity. The effect of scaling laws on other components was examined and it was concluded that the environments identified by the shaded numbers in Table 6-A could be dropped from further consideration.

Derivation of covariance between environment systematic variations and random variations was based upon environment prediction equations using the matrix approach described in Section 2.3. Table 6-D shows the starting point, which is a list of the weapon effects environment equations (for a fixed range) in terms of the analysis coefficients, load, site, and structural parameters. The exponents of the variables of the environment equations are summarized in matrix form in Table 6-E. In the table the environments are divided into PARE and CARE effects, with the PARE effects further subdivided into free field and structural effects. Table 6-F shows the sources of uncertainty for the sample problem and gives the K-factors for both random and systematic variations. In general, a K-factor is a parameter such that most random outcomes lie between K and 1/K times the best estimate value, $\bar{M}$. K-factors are exact descriptors of variation for log-normal distributions and approximate for others. A two-sigma K-factor corresponds to 2σ bounds on a log-normal distribution; i.e., 2-1/2% of outcomes greater than $K\bar{M}$ and 2-1/2% of random

Table 6-D. Weapon Effect Predictions

Weapon Effect Parameter	Weapon Effect Equation
E(1), d_z	$C_1\, p^{0.67} I^{0.5} L^{0.6} \rho^{-0.5} M^{-0.5}$
E(2), v_z	$C_2 p\, \rho^{-0.5} M^{-0.5}$
E(3), a_z	$C_3 C_2 p\, \rho^{-0.5} M^{-0.5} c\ (=C_3 v_z c)$
E(4), $V_{z\ max}$	$C_4 C_2 p\, \rho^{-0.5} M^{-0.5}\ (=C_4 v_z)$
E(5), $20\, d_z^{\ 2}/v_z$	$C_5 C_1^{\ 2} C_2^{\ -1} p^{0.33} I L^{1.2} \rho^{-0.5} M^{-0.5}$
E(6), d_H	$C_6 C_1 p^{0.67} I^{0.5} L^{0.6} \rho^{-0.5} M^{-0.5}\ (=C_6 d_z)$
E(7), v_H	$C_7 \rho^{-1} p^{0.5}$
E(8), $V_{H\ max}$	$C_8 C_7 \rho^{-1} p^{0.5}$
E(9), $1.5\, v_H^{\ 2}/d_H$	$C_9 C_6^{\ -1} C_1^{\ -1} \rho^{-1.5} p^{0.33} I^{-0.5} L^{-0.6} M^{0.5} \cdot C_7^{\ 2}$
E(10), $\sigma \cdot d_{rel}$	$C_{10} p^{1.8} M^{-1}$
E(11), Q	$C_{11} p^{1.37}$
E(12), a_{STR}(A/C)	$C_{12} p$
E(13), a_{STR}(SSS)	$C_{13} p$
E(14), Roof Load	$C_{14} p\, F_R^{\ -1} G_f^{\ 0.1}$
E(15), Wall Load	$C_{15} p\, F_W^{\ -1} G_f^{\ 0.1}$
E(16), Footing Load	$C_{16} p\, F_F^{\ -1} G_f^{\ -0.2}$
E(17), d_{CARE}	$C_{17} V^{1.33}$
E(18), v_{CARE}	$C_{18} V^{0.67} c$
E(19), Debris Depth	$C_{19} V^{1.6}$

Table 6-E. Matrix of Environment Variable Exponents

Analysis Coefficients (C_j)

		Environment Name	#	Free Field 1	2	3	4	5	6	7	8	9	10	11	[illegible]tural 12	[illegible]	4	15	16	CARE 17	18	19
PARE	FREE FIELD	d_z	1	1																		
PARE	FREE FIELD	v_z	2		1																	
PARE	FREE FIELD	a_z	3		1	1																
PARE	FREE FIELD	$V_{z\ max}$	4		1		1															
PARE	FREE FIELD	$20d_z^2/v_z$	5	2	-1			1														
PARE	FREE FIELD	d_H	6	1					1													
PARE	FREE FIELD	v_H	7							1												
PARE	FREE FIELD	$V_{H\ max}$	8							1	1											
PARE	FREE FIELD	$1.5\ v_H^2/d_H$	9	-1					-1	2		1										
PARE	FREE FIELD	$\sigma \cdot d_{rel}$	10										1									
PARE	FREE FIELD	Q	11											1								
PARE	STRUCTURE	a_{STR}(A/C)	12												1							
PARE	STRUCTURE	a_{STR}(SSS)	13													1						
PARE	STRUCTURE	Roof Load	14														1					
PARE	STRUCTURE	Wall Load	15															1				
PARE	STRUCTURE	Footing Load	16																1			
CARE		d_{CARE}	17																	1		
CARE		v_{CARE}	18																		1	
CARE		Debris Depth	19																			1

Underlying Variables

		Environment Name	#	Loads p	I		Site Properties c	M	ρ	L	Struct Properties F_R	F_W	F_F	G_f
PARE	FREE FIELD	d_z	1	0.67	0.5			-0.5	-0.5	0.6				
PARE	FREE FIELD	v_z	2	1				-0.5	-0.5					
PARE	FREE FIELD	a_z	3	1			1	-0.5	-0.5					
PARE	FREE FIELD	$V_{z\ max}$	4	1				-0.5	-0.5					
PARE	FREE FIELD	$20d_z^2/v_z$	5	0.33	1			-0.5	-0.5	1.2				
PARE	FREE FIELD	d_H	6	0.67	0.5			-0.5	-0.5	0.6				
PARE	FREE FIELD	v_H	7	0.5					-1					
PARE	FREE FIELD	$V_{H\ max}$	8	0.5					-1					
PARE	FREE FIELD	$1.5\ v_H^2/d_H$	9	0.33	-0.5			0.5	-1.5	-0.6				
PARE	FREE FIELD	$\sigma \cdot d_{rel}$	10	1.8				-1						
PARE	FREE FIELD	Q	11	1.37										
PARE	STRUCTURE	a_{STR}(A/C)	12	1										
PARE	STRUCTURE	a_{STR}(SSS)	13	1										
PARE	STRUCTURE	Roof Load	14	1							-1			0.1
PARE	STRUCTURE	Wall Load	15	1								-1		0.1
PARE	STRUCTURE	Footing Load	16	1									-1	-0.2
CARE		d_{CARE}	17			1.33								
CARE		v_{CARE}	18			0.67	1							
CARE		Debris Depth	19			1.6								

Table 6-F. Systematic and Random Variation of the Basic Parameters

Parameter	$2\sigma K$ Values		Parameter	$2\sigma K$ Values	
	Systematic	Random		Systematic	Random
C_1	2.0	1.0	M	3.0	5.0
C_2	1.2	1.0	ρ	1.1	1.1
C_3	1.7	1.2	L	1.3	1.5
C_4	1.3	1.0	p	1.2	1.2
C_5	1.5	1.0	I	1.2	1.2
C_6	1.7	1.3	c	1.5	1.7
C_7	2.0	1.0	V	1.6	1.2
C_8	1.3	1.0	F_R	1.2	1.1
C_9	1.3	1.0	F_W	1.2	1.1
C_{10}	1.4	1.0	F_F	1.2	1.1
C_{11}	1.5	1.05	G_f	2.0	2.0
C_{12}	1.5	1.0			
C_{13}	1.5	1.0			
C_{14}	1.3	1.0			
C_{15}	1.5	1.0			
C_{16}	1.5	1.0			
C_{17}	2.5	1.2			
C_{18}	2.5	1.2			
C_{19}	2.0	1.3			

outcomes less than $\overline{M}/K$. For this problem, 2σ bounds are used. Hence, the variances, σ_i^2, of the underlying sources of variation may be computed from the corresponding K-factors as follows:

$$\sigma_i^2 = \frac{(\ln K_i)^2}{4} \tag{6-4}$$

For this problem, it is assumed that the random variation of the analysis coefficients is small because the analysis technique is based on a single geology of interest and the random variation of the identified underlying variables will account for most, if not all, of the random variance. Therefore, several of the random K-factors are shown as unity in Table 6-F. On the other hand, the K-factors for systematic bias of the analysis coefficients are relatively large, because the trend lines of the prediction equations are not based on firm test and/or analysis data.

Correlation coefficients and K-factors for the materials, loads and structural parameters are shown in Table 6-G, including correlations between M, c and ρ; correlations between p and I; correlations between G and M; and correlations among the structural load factors, where M is confined modulus, c is peak stress wave velocity and ρ is density of the soil; p is pressure, I is impulse, G_f is foundation shear modulus of the soils, and the remaining parameters in the table are structural factors.

Table 6-G. Correlation Among Basic Parameters

Parameters		
i	j	ρ_{ij}
C_{17}	C_{18}	0.4
C_{12}	C_{13}	0.2
C_{14}	C_{16}	0.3
M	c	0.8
M	ρ	0.3
c	ρ	0.3
I	p	0.7
F_R	F_W	0.8
F_F	F_W	0.8
F_R	F_F	0.7
G_f	M	0.5

The random and systematic covariance matrices of the underlying variables (Σ_{vv}) can now be constructed from the K-factors and correlation coefficients. Then, from Equation (2-19), the corresponding environment covariance matrices can be computed. Table 6-H gives 2σ K-factors for the systematic and random variation of the environments. The covariance matrices are contained in the Input Summary, printed out with each FAST run. These are in Appendix A.

6.2 TRANSFER FUNCTIONS AND FRAGILITIES

The derivation of transfer functions and fragilities is best described together. Therefore, the discussion of both topics is given in this section. Two specific examples are used to illustrate in detail the derivation of transfer functions and fragility curves, namely, for the antenna probe and the shock suspension system structural failure. Derivations for other components, while not given, would follow a similar development.

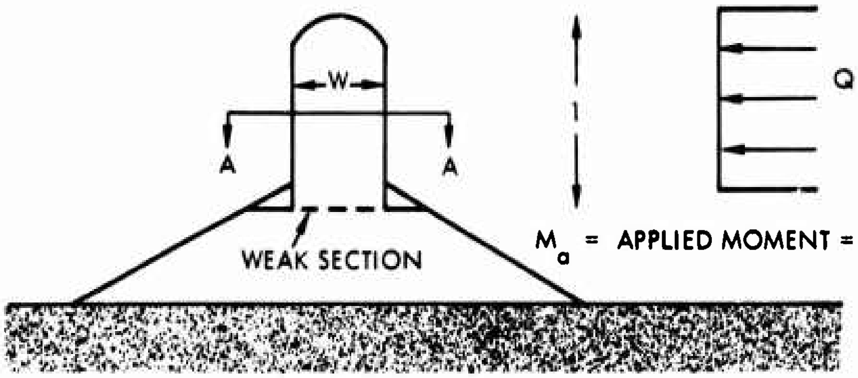
Sometimes fragilities are defined in a normalized fashion, by dividing the unnormalized fragility by the allowable load. In these instances, the unnormalized transfer function must also be divided by the allowable load (as shown in figures of Section 6.3). Fragility and transfer function data can be displayed graphically or given in terms of K-factors, whichever is most convenient.

6.2.1 Communications Antenna Probe - Fragility and Transfer Function

Figure 6-3 shows a cross-section of a communication antenna which has been designed for the hypothetical control facility. The probe, of width w and length l, is cylindrical as shown in Section A-A. The probe is vulnerable to dynamic pressure loading, which creates a moment, $M_a = c_d \, Q \, wl^2/2$, about the weak rectangular section shown, where c_d is the drag coefficient and Q is the dynamic pressure. The exterior of the probe is axisymmetric. Therefore, the applied moment, M_a, does not vary with azimuth. However, the allowable moment, $M_p = \sigma_a Z(\phi)$, is azimuth-sensitive because the plastic section modulus, $Z(\phi)$, is a function of azimuth, with the strongest section facing North, where σ_a is the allowable stress.

Table 6-H. K-Factors for Hostile Environments

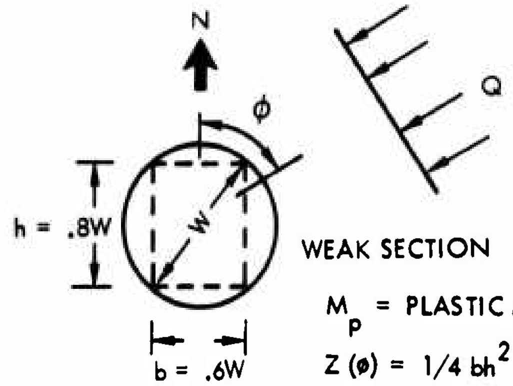
Environment		Systematic	Random
Number	Name	K_S	K_R
1	d_z	2.5	2.4
2	v_z	1.9	2.3
3	a_z	2.0	1.8
4	V_z	2.0	2.3
5	$20\, d_z^2/v_z$	5.0	2.6
6	d_H	2.9	2.5
7	v_H	2.0	1.1
8	V_H	2.1	1.1
9	$1.5\, v_H^2/d_H$	5.7	2.3
10	$\sigma \cdot d_{rel}$	3.3	5.2
11	Q	1.6	1.3
12	a_{STR}(A/C)	1.6	1.2
13	a_{STR}(SSS)	1.6	1.2
14	Equivalent Static Roof Load	1.4	1.2
15	Equivalent Static Wall Load	1.6	1.3
16	Equivalent Static Footing Load	1.6	1.3
17	d_{CARE}	3.0	1.3
18	v_{CARE}	2.9	1.8
19	Debris Depth	2.8	1.5



M_a = APPLIED MOMENT = $c_d Q \frac{wl^2}{2}$

SIDE VIEW

TOP VIEW - SECTION AA



WEAK SECTION

M_p = PLASTIC MOMENT = $\sigma_a Z(\phi)$

$Z(\phi) = 1/4\ bh^2\ (1-1/3\,\eta^2)/\cos\phi \quad 0 \le \phi \le 37^\circ$

WHERE $\eta = \sqrt{\left(\frac{b}{h}\cot\phi\right)^2 + 3} - \frac{b}{h}\cot\phi$

$Z(\phi) = 1/4\ hb^2\ \left(1 - \frac{\eta^2}{3}\right) / \cos(90^\circ - \phi) \quad 37^\circ \le \phi \le 90^\circ$

WHERE $\eta = \sqrt{\left(\frac{h}{b}\cot(90^\circ - \phi)\right)^2 + 3} - \frac{h}{b}\cot(90^\circ - \phi)$

$AZ(\phi) = \frac{Z_{NORTH}}{Z(\phi)}$

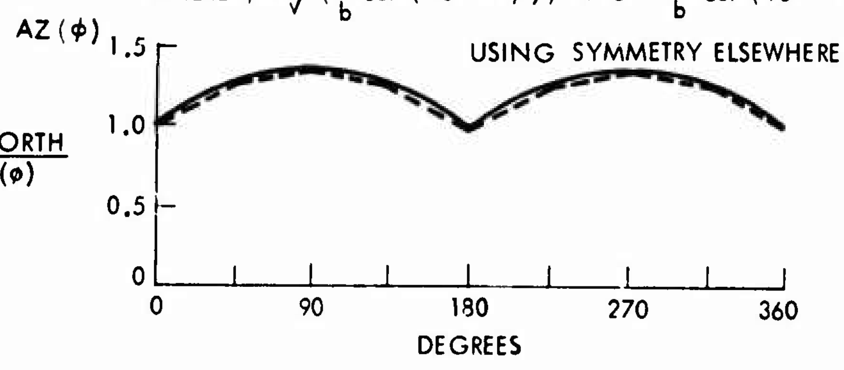


Figure 6-3. Antenna Structure Description (Plain View and Elevation)

The critical response parameter is the ratio of applied moment to allowable moment,

$$R = \frac{c_d w l^2}{2Z(\phi)} \frac{Q}{\sigma_a} = \frac{Z_{NORTH}}{Z(\phi)} \cdot \frac{c_d w l^2}{2Z_{NORTH}} \frac{Q}{\sigma_a} \qquad (6\text{-}5)$$

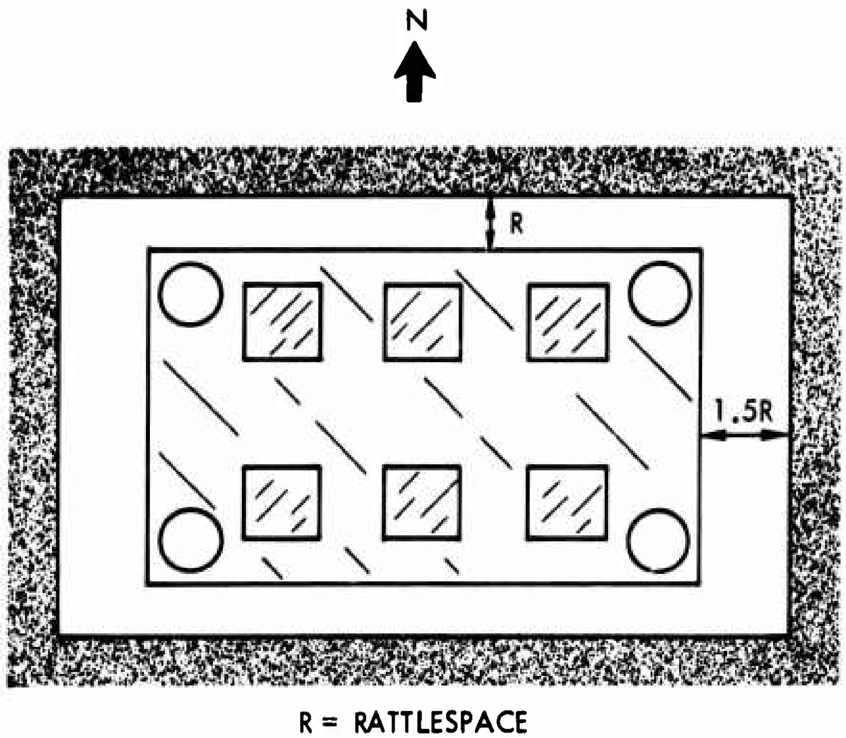
where Z_{NORTH} is the plastic section modulus versus north attack, and $Z_{NORTH}/Z(\phi)$ is the azimuth sensitivity which is incorporated into the transfer function. Note that the dashed lines, which are input to FAST, accurately represent the continuous function both in the mean and in the variance of the transfer function from 0^o to 360^o.

Azimuth sensitivity also can be caused by non-uniform rattlespace. Obviously, non-uniform rattlespace is not good, but sometimes it cannot be avoided. Figure 6-4 shows a plan view of the rattlespace and the azimuth sensitivity which could result for pure translational motion. In practice, pure translational motions are unlikely and cusps on azimuth sensitivity curves will be rounded by random variation.

6.2.2 S^3 Structural Failure Fragility and Transfer Function

The following discussion considers only the vertical input motions and responses, but similar concepts can be applied to other directions. The shock suspension system (SSS) will fail structurally when the floor motion exceeds the length of the allowable excursion for the rod which attaches the floor to the isolators. The isolators are carefully calibrated to fail all at once or not at all; hence, one displacement fragility curve applies to all isolators.

The vertical natural frequency of the S^3 is 2 cps. In some situations 2 cps corresponds to the constant displacement portion of the shock spectrum (see Figure 6-5). In other instances, 2 cps intersects the constant velocity portion, and sometimes the displacement amplification region of the spectrum covers 2 cps. Transfer function uncertainties are used to



$$AZ\ (\phi) = \frac{\text{RATTLESPACE } (\phi)}{\text{NORTH RATTLESPACE}}$$

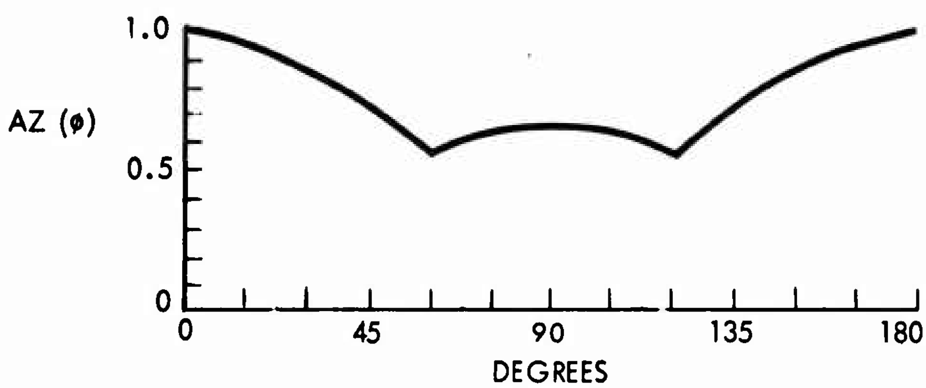


Figure 6-4. SSS Horizontal Rattlespace Description (Plan View)

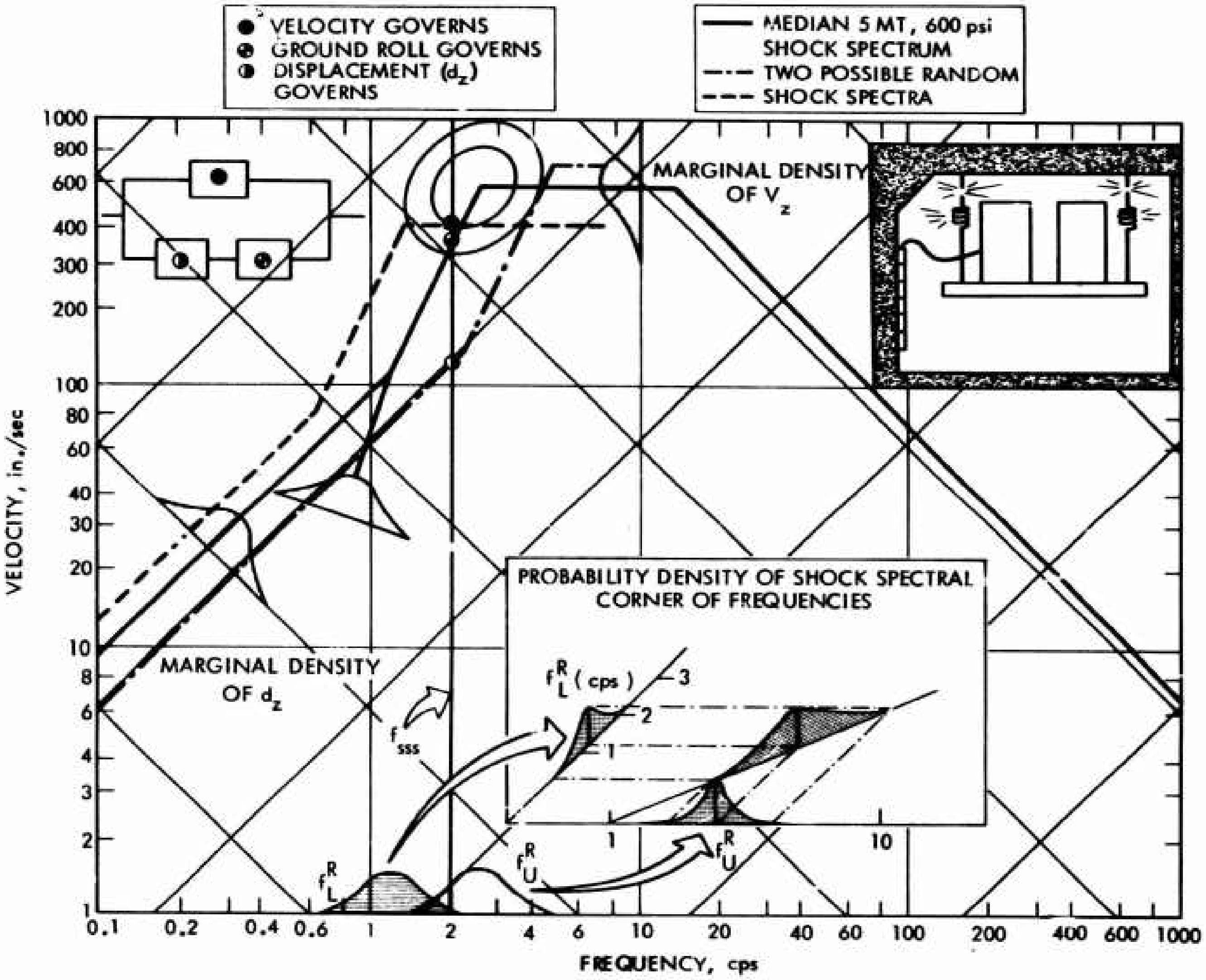


Figure 6-5. SSS (Section) and Shock Spectra, with Bivariate Distribution of Shock Spectra Corner Frequencies, F_U and F_L

allow for the limitations of the data base and for the straight-line simplifications of this possible cause of excessive excursion. The S^3 shock spectrum value may be any of the following:

Portion of Shock Spectrum	Peak Free Field Value	Shock Spectrum Value		Applicable Transfer Function
		In Terms of Displacement	In Terms of Velocity	
Constant Displacement	d	D	$V/2\pi f$	1.0
Displacement Amplification	d	$20f\ d^2/v$		2.0
Constant Velocity	v	$D \times 2\pi f$	V	$1/4\pi$

The logic of S^3 failure is series-parallel. The series portion implies that the S^3 will fail if the larger of the displacement amplification and the peak free field vertical displacement exceeds the available stroke length. The parallel portion implies that the subsystem will survive if either a) the relative displacement at 2 cps on the shock spectral velocity line is less than the stroke length or b) d_z is less than the stroke length of the S^3.

Figure 6-5 shows the effects of random variation on the shock spectra. The elliptical contours show how one of the shock spectral corners might vary. Also, shown are the marginal densities of D, V and $20d^2/v$ which are all computed from only 2 free field variables. Systematic variations among these variables are not linearly dependent in log space due to variation of analysis constant (shown in Table 6-F), which is the systematic variation K factor for the fourth analysis coefficient.

The cut-out in Figure 6-5 shows the random bivariate distribution of f_L^R versus f_U^R, which are the random parts of the shock spectra break frequencies. Note that the fourth analysis constant in Table 6-F has a random K factor, $K_R = 1$. This implies that f_L/f_U is constant and the lower corner frequency is linearly dependent on the upper corner

frequency (in log space). This will result in a singular covariance matrix for random environments. But this does not matter since FAST can decompose and use singular covariance matrices for random number generation in the simulation.

6.3 SYSTEM NETWORK

The preliminary baseline form for the system network is shown in Figure 6-6. This network is the result of a good deal of pruning of larger networks based upon knowledge of environment distributions and component criticality. Each of the subsystem networks, together with fragility and transfer function definitions, is shown in Figures 6-7 to 6-14

One point not resolved in the preliminary system network, concerned the air conditioner. Two alternatives were postulated for the design of this subsystem; one having the fragility shown in Figure 6-14, the other having a superhard design. Resolution of this decision was based upon the FAST survivability statistics.

6.4 SURVIVABILITY CALCULATION RESULTS

To illustrate use of the FAST methodology and code, assume a requirement that the system have a probability of survival of at least 0.8 with 50% confidence when subjected to a surface-burst nuclear attack involving a single 600 psi overpressure pulse from a 5-megaton weapon. The inputs derived in Section 6.1 - 6.3 are assumed to be for a baseline system, which is the result of an initial design effort. The FAST technique is used for evaluating the baseline system design and identifying how best to modify it to meet the system requirements.

FAST survivability statistics used to evaluate the baseline system show that it fails to meet the survivability requirement. The calculation also provides information on how the baseline system design could be improved because critical subsystems and components are identified. A sensitivity analysis can be done, employing a steepest ascent approach to identify the most cost effective way to strengthen the system design to meet the system requirement. The steepest-ascent technique is not an inherent part of the FAST methodology, but is indicative of how the FAST results can be employed to provide data for management decisions.

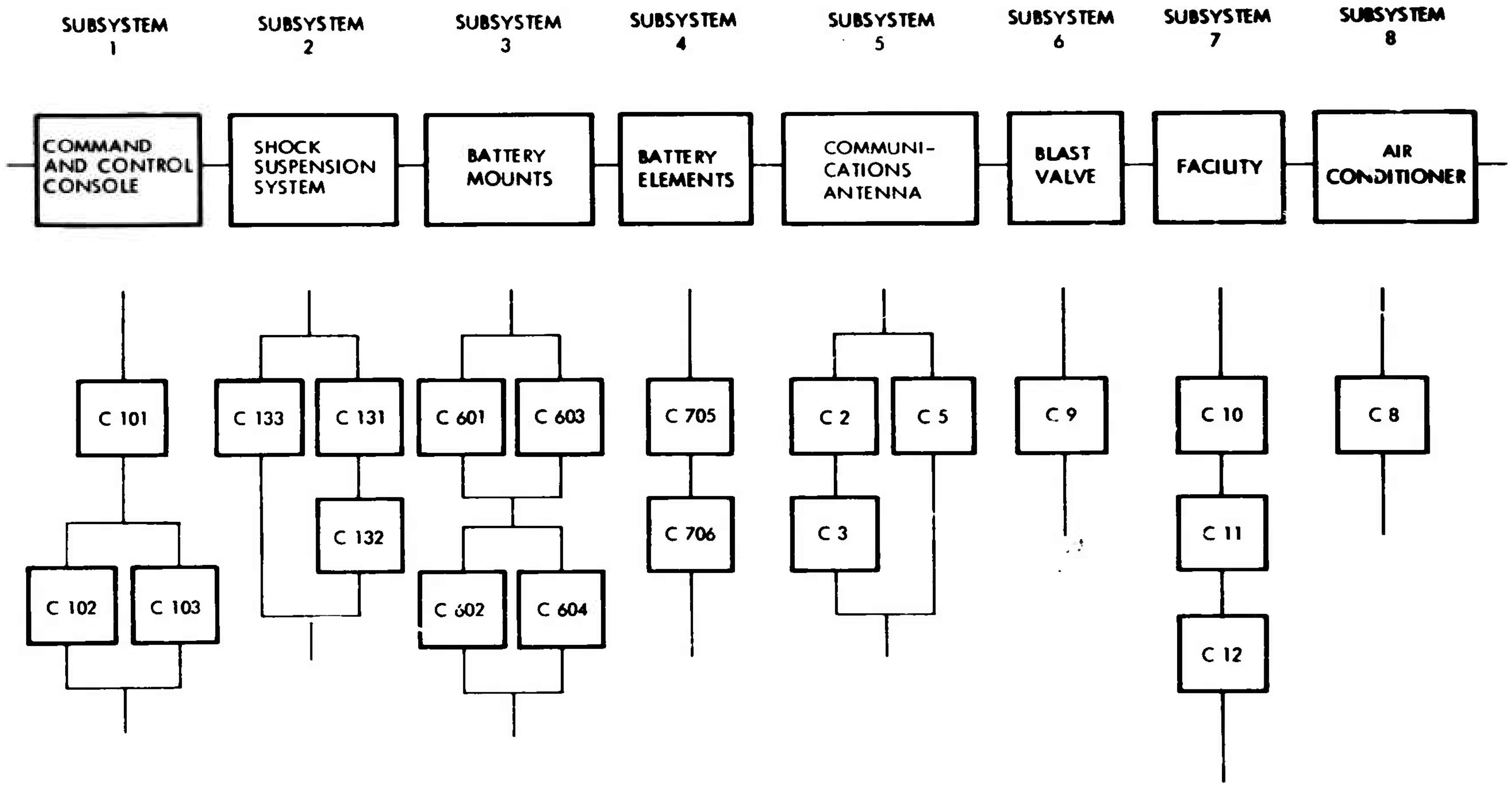
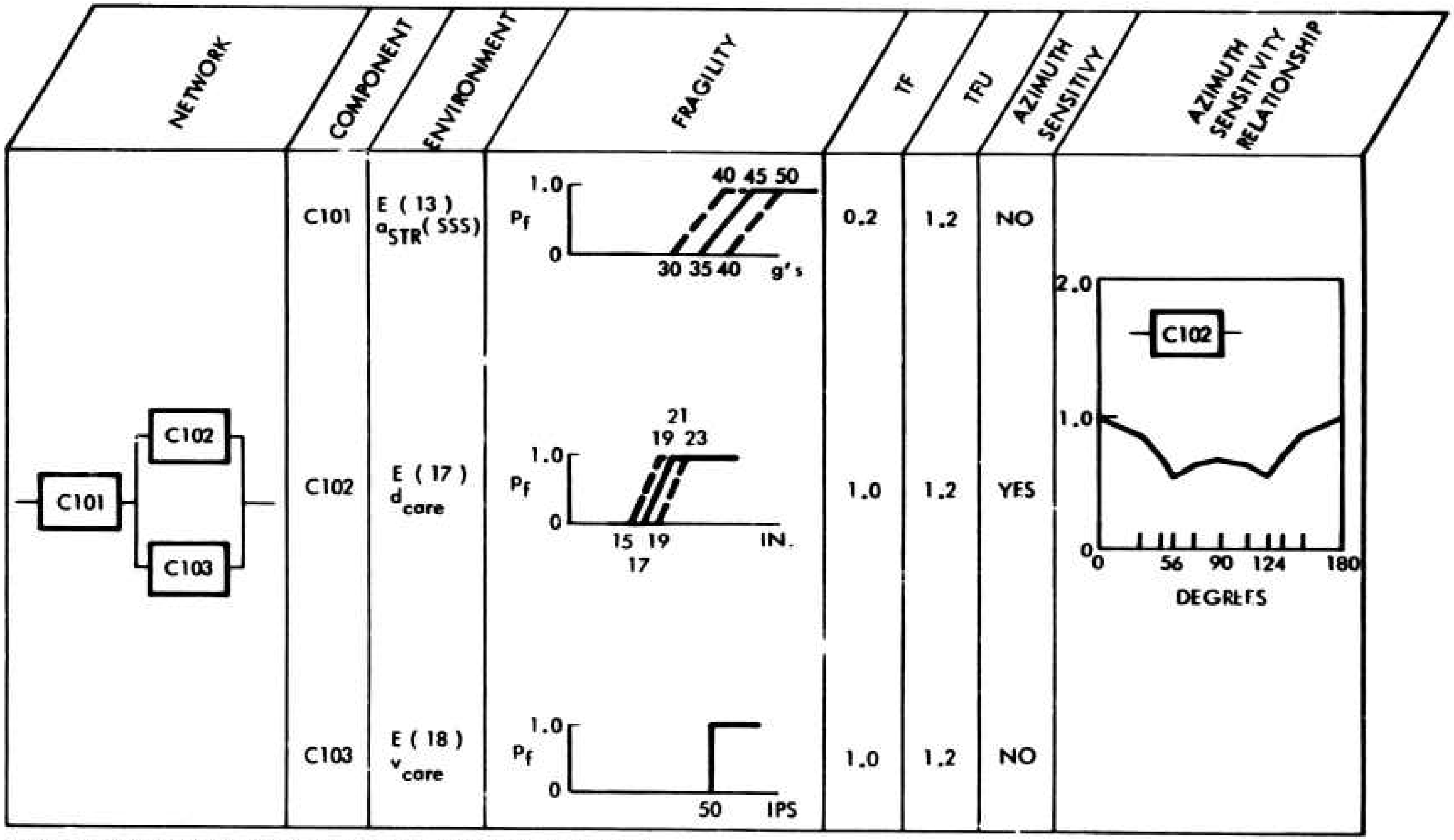


Figure 6-6. System Network



NOTE: COMPONENTS C102 AND C103 ARE NOT INTENDED TO MODEL THE SHOCK SPECTRA CORNER

Figure 6-7. Command Control Console

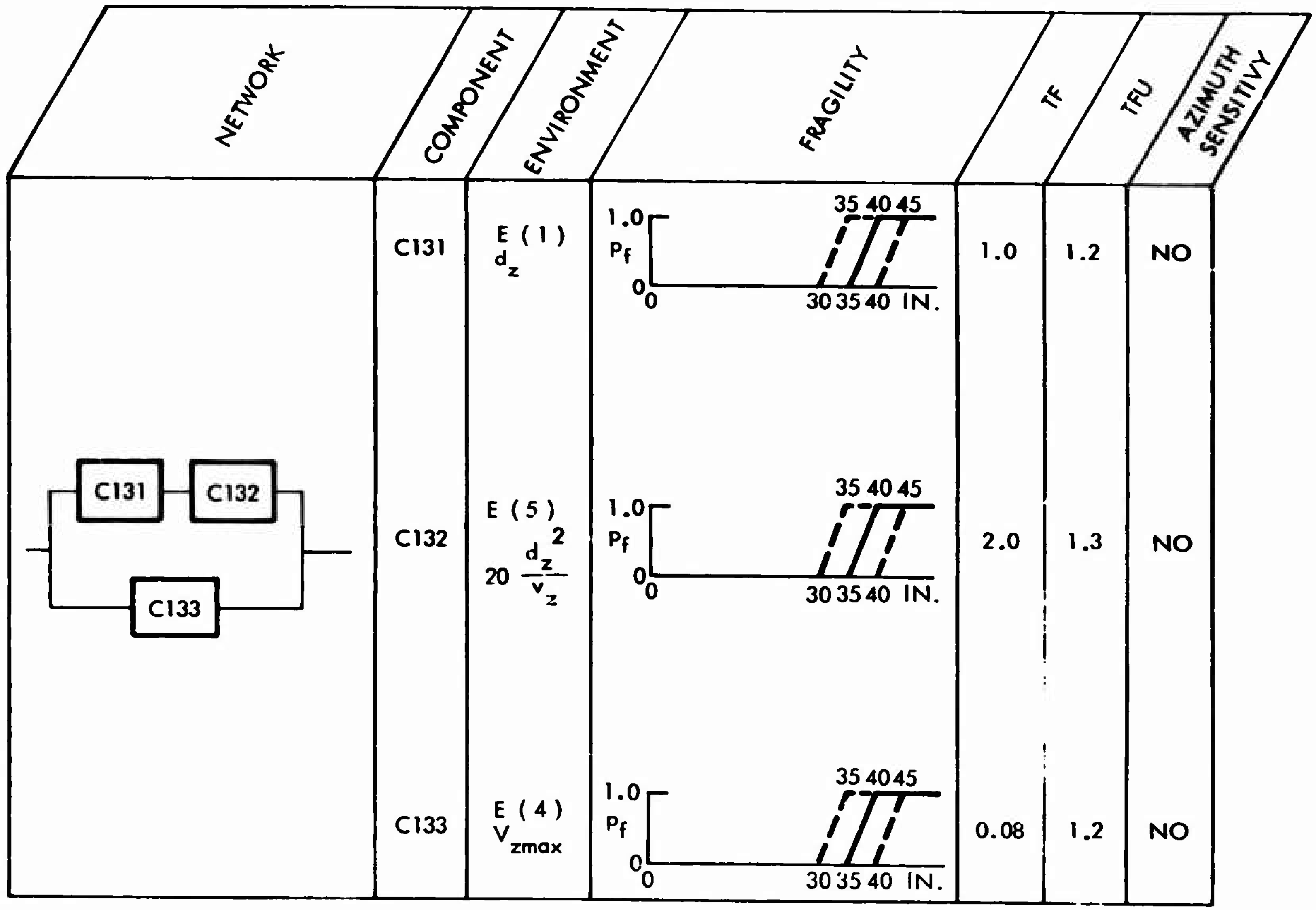


Figure 6-8. Shock Suspension System

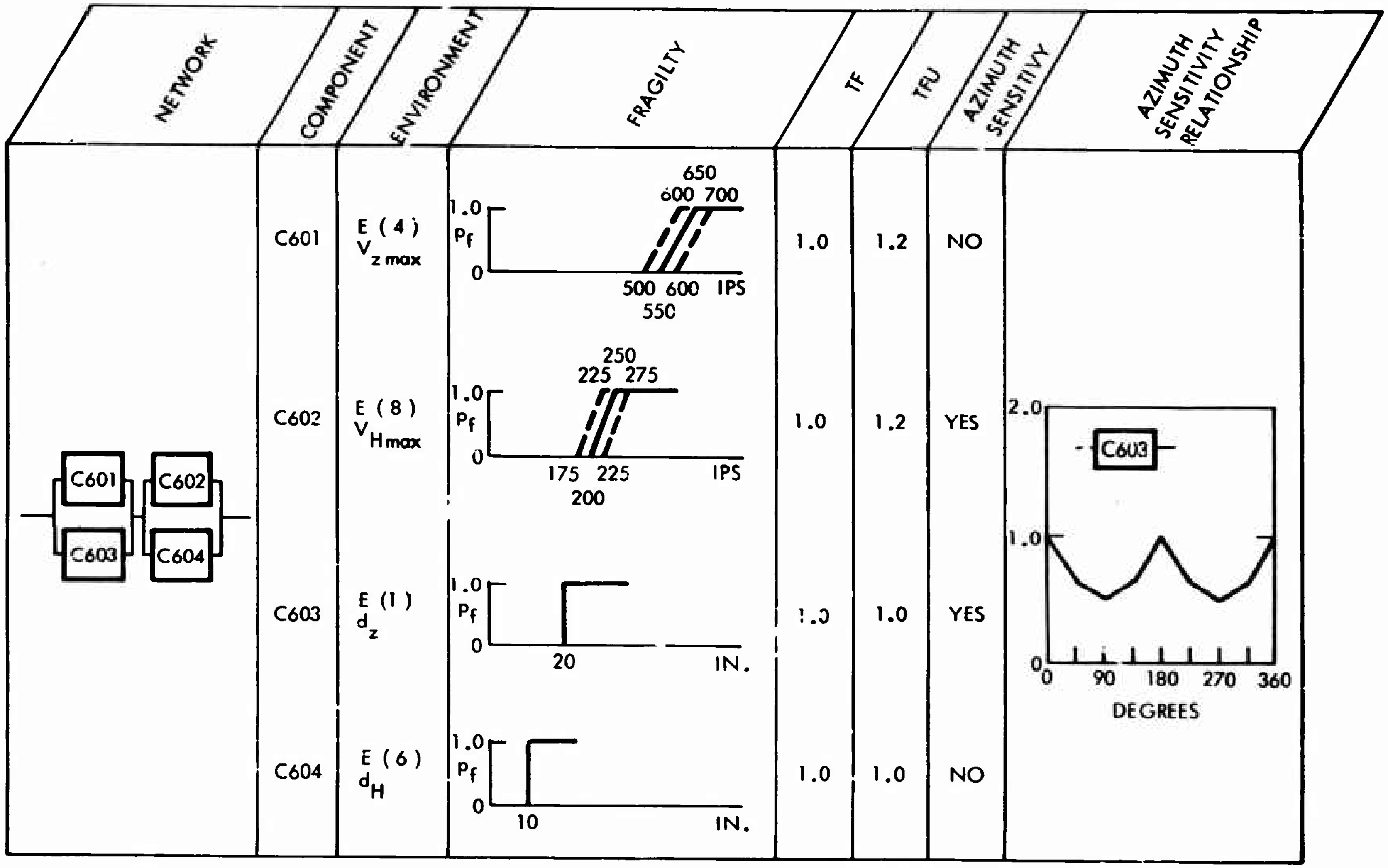


Figure 6-9. Battery Supports

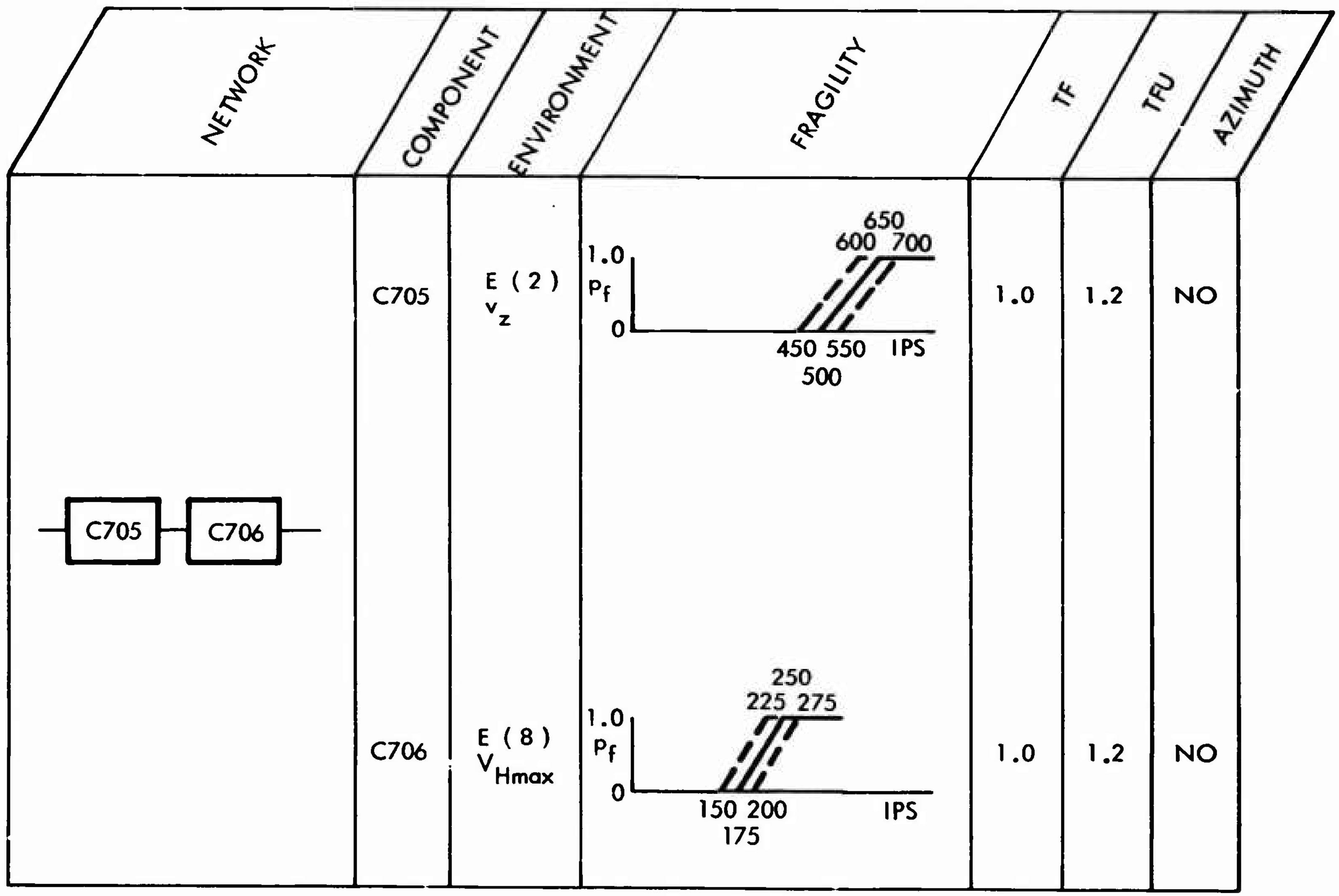


Figure 6-10. Batt :ry Elements

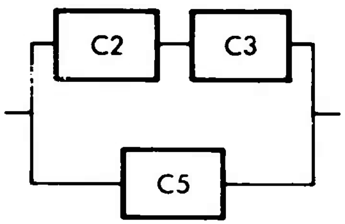
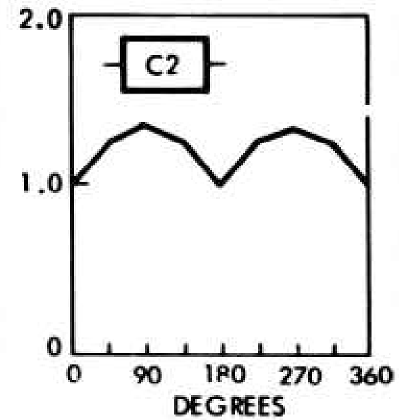
NETWORK	COMPONENT	ENVIRONMENT	MEDIAN FRAGILITY LEVEL	TF	TFU	K_R	K_S	AZIMUTH	AZIMUTH SENSITIVITY
	C2	E (11) Q psi	1.0	0.0004	1.3	1.05	1.3	YES	
	C3	E (19) DEBRIS IN.	1.0	0.04	1.4	1.05	2.0	NO	
	C5	E (10) $\sigma\ d_{rel}$ psi - IN.	1.0	0.0002	1.3	1.2	1.3	NO	

Figure 6-11. Communication Antenna System

NETWORK	COMPONENT	ENVIRONMENT	MEDIAN FRAGILITY LEVEL	TF	TFU	K_R	K_U	AZIMUTH SENSITIVITY
C9	C9	E (0) ΔP psi	1.0	0.001	1.2	1.2	1.2	NO

Figure 6-12. Blast Valve

NETWORK	COMPONENT	ENVIRONMENT	MEDIAN FRAGILITY LEVEL	TF	TFU	K_R	K_U	AZIMUTH SENSITIVITY
C10 — C11 — C12	C10	E (14) ROOF LOAD psi	1.0	0.0015	1.1	1.2	1.2	NO
	C11	E (15) WALL LOAD psi	1.0	0.00135	1.2	1.3	1.2	NO
	C12	E (16) FOOTING LOAD psi	1.0	0.0012	1.3	1.4	1.2	NO

Figure 6-13. Facility

NETWORK	COMPONENT	ENVIRONMENT	MEDIAN FRAGILITY LEVEL	TF	TFU	K_R	K_U	AZIMUTH SENSITIVITY
C8	C8	E (12) a_{STR} (A/C) g's	1.0	0.01	1.4	1.2	1.3	NO

Figure 6-14. Air Conditioner

The sensitivity analysis leads to a configuration satisfying the system requirement which is then used to illustrate the effects of weapon yield and height-of-burst perturbations on the survivability statistics.

6.4.1 Evaluation of the Baseline System

Evaluation of the baseline system was accomplished by means of the first computer run, which is reproduced in Appendix A. The inputs for this run were derived in Sections 6.1 through 6.3.

An overview of the results from the first run is given in Figure 6-15. The system probability of survival is 0.05, much less than the required 0.8. Four of the subsystems have a probability of essentially 1.00. Thus, to achieve the required system hardness some combination of the other four subsystems must be hardened. At this point, it was decided to use the superhard version of subsystem 8, the air conditioner. Therefore, this subsystem was deleted from further analysis.

Several aspects of baseline system survivability statistics are plotted in Figure 6-16. The system probability of survival systematic variation distribution (10, 50 and 90% levels) is shown at the upper left as a function of overpressure.

In the upper right is plotted the median value of the probability of survival of the subsystems as functions of overpressure. As can be seen, the critical subsystems are the command and control console C^3 (SS1), shock suspension system (SS2), and the facility (SS7).

The other three diagrams break down the three critical subsystems. The median probability of survival of the subsystem, together with the median probability of survival of the components, are shown as functions of overpressure.

Evaluation of the subsystem survivability statistics in Figure 6-16 identified three areas where hardening is required. The first is component C101 in Subsystem 1, which causes almost all of the failure of this subsystem. The second area is Subsystem 2, the Shock Suspension System, where it is natural to harden the three fragilities all in the same proportion by redesigning the subsystem. The third area is the facility, Subsystem 7. Here also, it is natural to improve all fragilities in

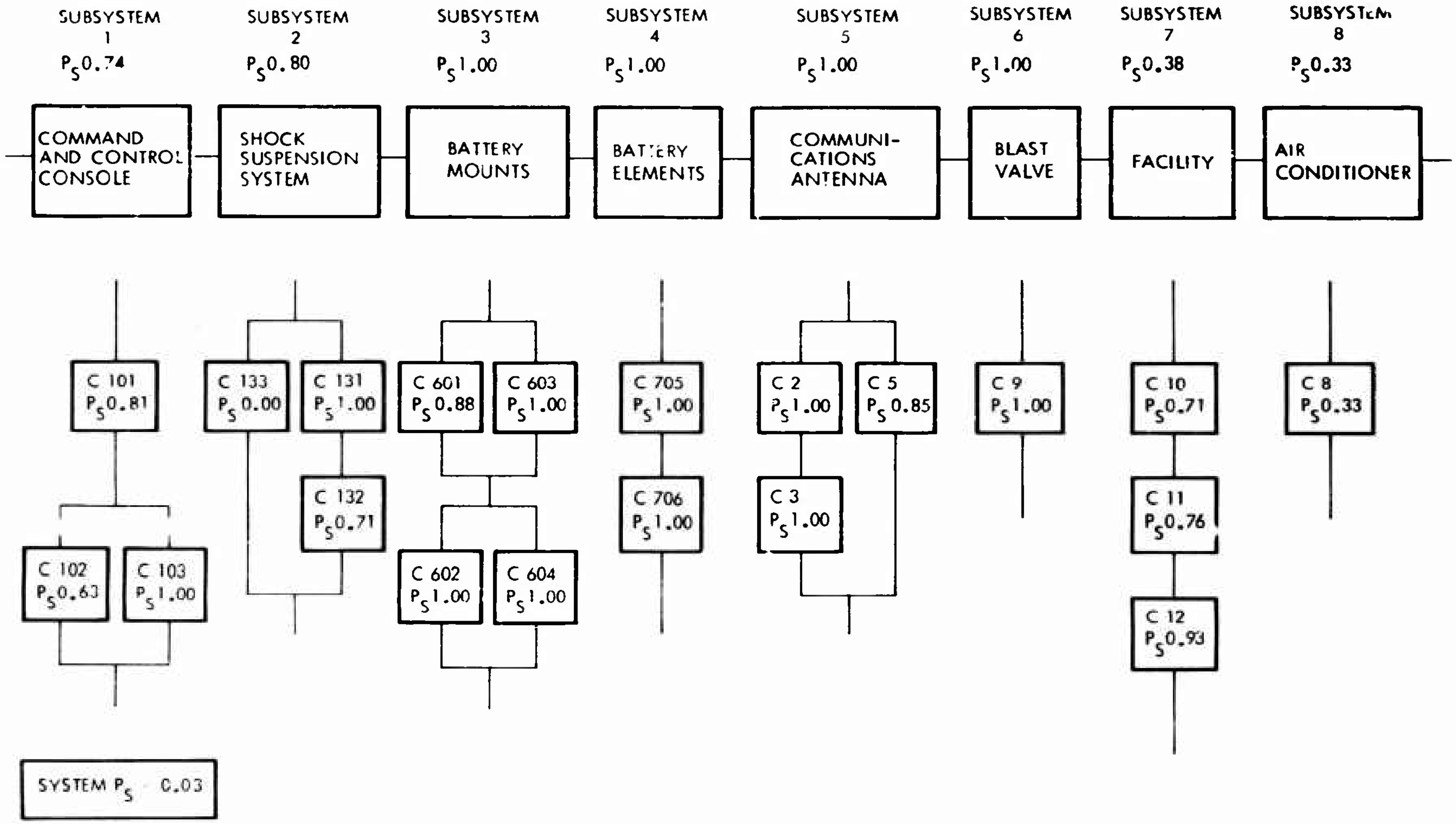


Figure 6-15. System Network and Baseline Probabilities of Survival (600 psi)

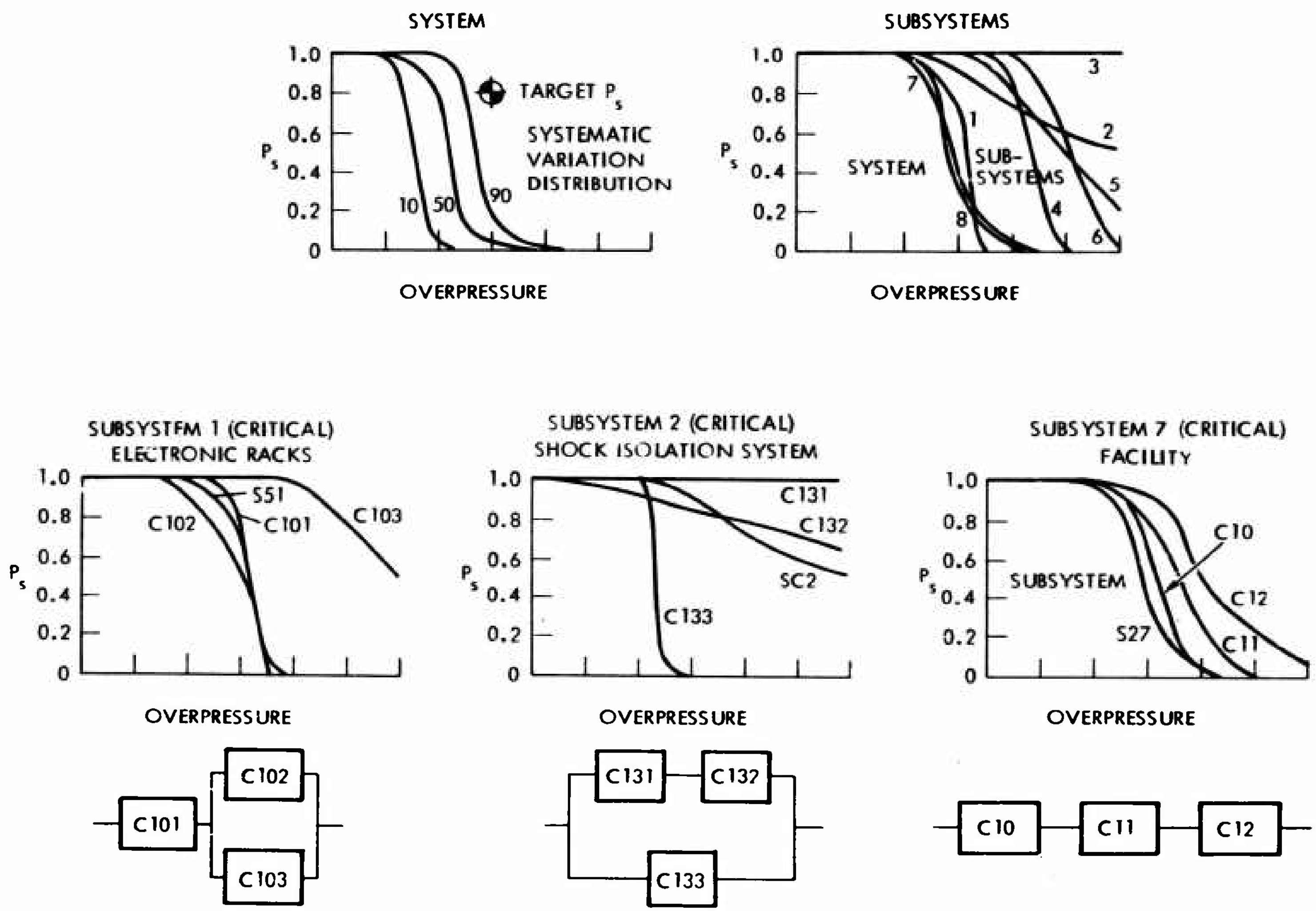


Figure 6-16. Baseline System Survivability Statistics

proportion because this is accomplished by using thicker concrete. Of course, in the case of a real system, other factors should also be considered before making a system change, such as how well the inputs for the critical subsystems are known, and whether or not further study in these areas might reduce the apparent criticality of these subsystems.

At this point, three areas have been identified requiring hardening to achieve the system survivability requirements. In the next section, these are treated as three variables and a steepest-ascent approach is employed to accomplish system hardening trade analysis.

6.4.2 System Hardening Trade Analysis

Once one has the three variables with the predominate effect on system hardness, the next step is to perform trade analysis and find a system design satisfying the survivability requirement. We shall neglect such possibly important considerations as system costs, accuracy of the data base, and confidence in achieving technical objectives. These factors would have to be considered in an actual situation but are not necessary to illustrate the concept. We will employ an optimization technique known as the method of steepest ascent to illustrate how such an optimization technique can be used with FAST.

The method of steepest ascent uses a statistically-designed experiment to find how the independent variables (three hardness variables in our case) affect the dependent variable (system survivability in our case). This information is then used to compute the way to simultaneously change the independent variables (called the direction of steepest ascent) to maximize the rate of increase of the dependent variable. System configurations along this line are evaluated to explore the survivability response. Typically, several cycles of experimental designs and steepest-ascent evaluations are employed to account for non-linearities and other contingencies. We shall limit our efforts to the first cycle, settling for the first system identified which satisfies the survivability requirement.

The experimental design is shown in Table 6-I. Two hardness levels are used for each of the variables, the baseline hardness and a higher value by 8 percent for the command and communication console, 15 for the suspension system, and 8 for the facility, respectively. Since there are

Table 6-I. Experimental Design Formulation and Results

System	Fragility Multipliers			System Probabilities Of Survival (Fast Output) 500 psi	Effect Name
	Command And Control Console C101	Shock Suspension System C131 C132 C133	Facility C10 C11 C12		
S1	1.00	1.00	1.00	0.35	Mean
S2	1.08	1.00	1.00	0.39	C101
S3	1.00	1.15	1.00	0.41	SIS
S4	1.08	1.15	1.00	0.46	C101 x SIS
S5	1.00	1.00	1.08	0.44	FAC
S6	1.08	1.00	1.08	0.49	C101 x FAC
S7	1.00	1.15	1.08	0.50	SIS x FAC
S8	1.08	1.15	1.08	0.57	C101 x SIS x FAC

*Data from 500 psi was used at this point in the analysis since the probabilities of survival at 600 psi were quite low.

two levels for each of the three variables, the total number of combinations is $2^3 = 8$, so this is the number of systems defined and evaluated by the FAST code in the experimental design.

The input section of the FAST printout for this computer run is reproduced as Run 2 in Appendix A. The probabilities of survival for the eight system configurations are shown in Column 5 of Table 6-I, together with the statistical analysis of the data.

From the data in Table 6-I, the direction of steepest ascent can be established. This was done, and approximately equally spaced points were selected along the line as shown in Table 6-J which defines 5 system configurations along the line of steepest ascent. Along this line, if fragility C101 is increased 25%, then the shock suspension system fragilities are increased 55% and the facility fragilities are increased 30%.

The final column of Table 6-J gives probabilities of survival for the five system configurations. As can be seen, systems SYS15 and SYS16 exceed the required 80% probability of survival. In the evaluation of a real system, many more tradeoff calculations would be done, to fine-tune the design. But for the purpose of this sample problem, system SYS16 will be selected as the final configuration. This system configuration is used in the calculations of the next section.

6.4.3 Yield and HOB Variations

The purpose of this section is to demonstrate the use of FAST code for calculations involving changes of weapon yield and height of burst (HOB). To do this, the "optimized" system obtained in the last section was used. Four combinations of yield and HOB were employed as follows:

		YIELD, MT	
		0.5	5.0
HOB, $\frac{\text{K ft}}{(\text{W/1 MT})^{1/3}}$	0	X	X
	1.5	X	X

Table 6-J. Results Along "Line of Steepest Ascent"

System	Fragility Multipliers			System Probabilities Of Survival (Fast Output) 600 psi
	Command And Control Console C101	Shock Suspension System C131 C132 C133	Facility C10 C11 C12	
SYS 12	1.04	1.08	1.04	0.17
SYS 13	1.25	1.55	1.30	0.59
SYS 14	1.50	2.10	1.60	0.75
SYS 15	1.75	2.65	1.90	0.80
SYS 16	2.00	3.20	2.20	0.80

The input section of the printout for this computer run is reproduced in Appendix A. The survivability statistics for the four cases are plotted in Figure 6-17.

At a scaled HOB of 1500 ft/(W/1 MT)$^{1/3}$, air-induced effects on the sample system are virtually the same as for a surface burst. However, the pressure-range relation in FAST has made the correction for the ranges at which the HOB pressures are obtained. Cratering effects, on the other hand, are drastically reduced. Crater volume scaling $E = E_o \exp(-\lambda HOB/W^{1/3})$ accounts for almost all of the reduction, while the slightly larger scaled range for the specific pressure levels at a given HOB decreases the crater induced effects by a negligible amount. Similar scaling can be applied for ranges where HOB/R $\leq$ 0.4. However, this scaling procedure breaks down as HOB approaches the optimal HOB for a given overpressure. Other scaling procedures based on site response to changing pressure-time histories are known to the weapons effects community. These scaling laws can be modeled in FAST by modifying environment and transfer function inputs.

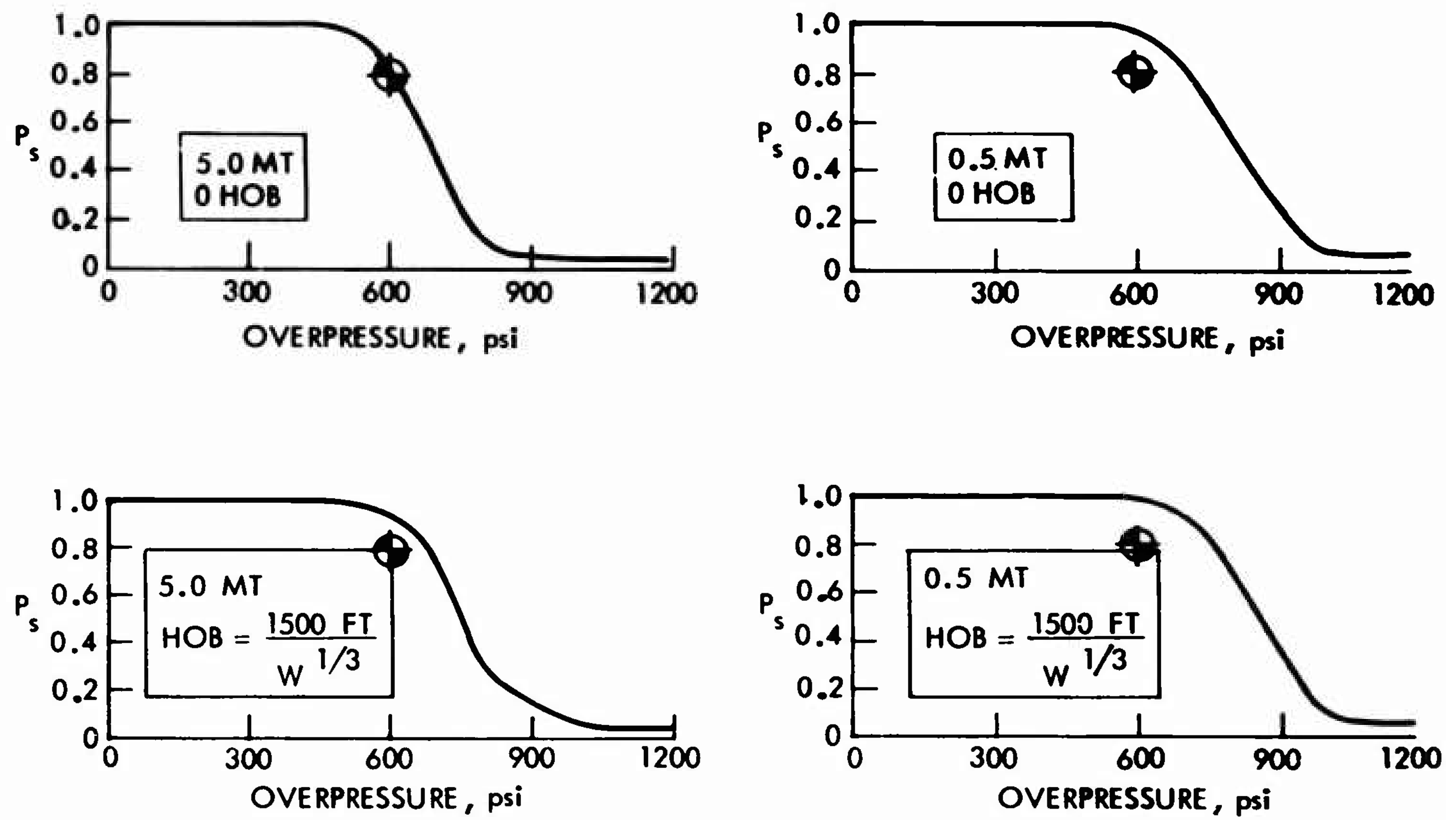


Figure 6-17. Survivability Statistics for "Optimized System for Weapon Yield and HOB Variations"

7.0 REFERENCES

1. T. W. Anderson, An Introduction to Multivariate Statistical Analysis. John Wiley & Sons, Inc., New York, 1958.

2. Gerald J. Hahn & Samuel S. Shapiro, Statistical Models in Engineering. John Wiley & Sons, Inc., New York, 1967.

3. H. L. Brode, "Height of Burst Effects at High Overpressures." The Rand Corporation, Santa Monica, California, 1970.

4. D. K. Faddeev and V. N. Faddeeva, Computational Methods of Linear Algebra. Translated by Robert C. Williams. W. H. Freeman and Company, San Francisco, 1963.